pom's

La revue francophone des utilisateurs de l'Apple

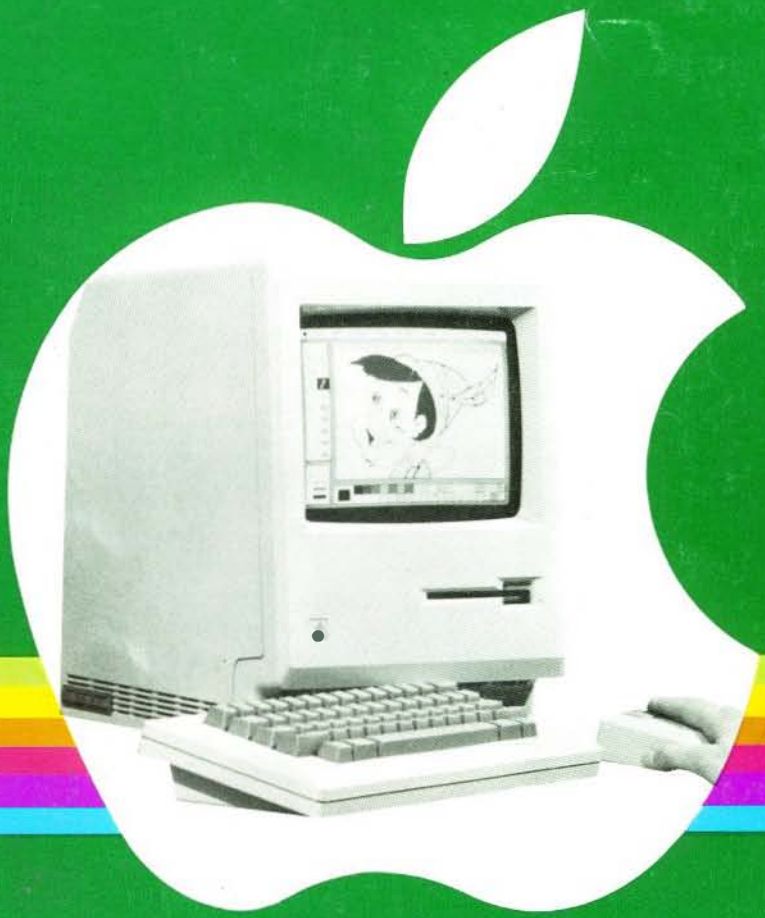
Faites parler et chanter votre Apple II

Des POKES à gogo

Tris rapides de tableaux

Initiation à l'assembleur

Le Macintosh est arrivé !



NUMÉRO 11 • PRIX 40 F

M2366-11-40 F

ISSN : 0294-6068

NE PERDEZ PLUS VOTRE TEMPS

DISQUETTEZ VOUS

14 LIVRES ET LEUR DISQUETTE POUR VOTRE APPLE II

POUR APPLE II, II PLUS, IIe

VISICALC SUR APPLE II • MATHEMATIQUES ET STATISTIQUES • LA PAIE ET SES ANNEXES
MODELES PRATIQUES DE DECISION (2 TOMES) • LE BASIC ET L'ECOLE (2 TOMES) •
METHODES DE CALCUL NUMERIQUE (2 TOMES) • BIBLIOTHEQUE SCIENTIFIQUE EN PASCAL
• NOUVELLE COMPTABILITE SUR APPLE II (TOME 1) • MODELES D'EXPRESSION GRAPHIQUE
• OUTILS FINANCIERS ET COMPTABLES • L'APPLE ET SES FICHIERS

POUR TOUT RENSEIGNEMENT TELEPHONEZ AU (6) 006 44 35
OU ECRIVEZ A P.S.I. DIFFUSION BP 86 77402 LAGNY/S/MARNE CEDEX

pom's n°11

Sommaire

	Page	Langage *	Niveau * *
Editorial par Hervé Thiriez	5		
Le Bavard par Gilles Chaygneaud-Dupuy	7	B-A	T
Une astuce pour supprimer les REMs par Guy d'Herbemont	9	B	T
Apprentissage de l'assembleur par Gérard Michel	10	A	D-T
Des POKEs à gogo par Roland Jost	15		P-T
Un programme de menu par Gilles Mauffrey	18	B-A	M-T
Les logiciels de traitement de texte par Guy Lapautre	20		T
Gutenberg à l'essai par Guy Lapautre	21		T
Pommesoft par Philippe Faure	24	B-A	P-T
Comparaison de programmes en langage-machine par Denis Sureau	33	A	P-T
Deux produits nouveaux pour les lecteurs de Pom's	37		
L'obtention de l'extremum absolu des fonctions de plusieurs variables par A. Guillez et Y. Cherruault	40	B	P
Un menu à la carte par Michel Crimont	45	P	T
Tri rapide en assembleur par Gérard Rigaud	47	B-A	P-T
Effets stroboscopiques par Olivier Herz	52	B-A	T
MacArticle par Daniel Garric	53		T
Magic Window II à l'essai par Alexandre Avrane	54		T
Comparaison de programmes Applesoft par Gérard Michel	57	B-A	P-T
Bibliographie par Alexandre Duback	61		
Micro-informations par Guy Lapautre	63		
Un « Type Ahead » en Applesoft par Olivier Herz	65	B-A	P-T
Courrier des lecteurs par Olivier Herz	68		
Tracé de courbes en conversationnel par Gilles Mauffrey	71	B-A	M-T
Trucs et astuces	14-46-59-66		

* Langage : B(asic) - A(ssembleur) - P(asca)). (B) signifie : relatif au BASIC.
** Niveau : D(ébutant) - M(oyen) - P(rofessionnel) - T(ous).
P-T signifie : programme utilisable par les débutants, mais dont la compréhension est de niveau "Professionnel".

Les annonceurs

ACER MICRO : p. 6 – **APPLE** : p. 38-39 – **B.F.I.** : p. 62 – **SERVICE CALVADOS** : p. 30-31-32 – **CEDIC-NATHAN** : p. 75 – **HELLO** : p. 73 – **IMAGOL** : p. 68 – **MICRO-PERIPH** : p. 56 – **L'ORDINATEUR INDIVIDUEL** : p. 60 – **PENTASONIC** : p. 76 – **P.S.I. Diffusion** : p. 2 – **TELECOMPO** : p. 34 – **SATELCOM** : p. 26 – **SICOB** : p. 4 – **SYBEX** : p. 70

Éditions MEV - 49, rue Lamartine - 78000 Versailles

Directeur de la publication : Hervé Thiriez. Imprimerie Rosay, 94300 Vincennes. Imprimé en France. Dépôt légal : 1er trimestre 1984.

SPECIAL SICOB UN SICOB TRES SPECIAL.

14-19 MAI
CNIT-PARIS
LA DEFENSE

PROGICIELS,
MINI, MICRO-ORDINATEURS
(JOURNÉES GRAND PUBLIC : 18 ET 19 MAI)

Information : SICOB (1) 261.52.42 – 4, place de Valois - 75001 Paris

Spécial SICOB

A 319

PUBLICIS

Editorial

Dès son lancement, Pom's a promis le passage en bimestriel dès que le nombre d'abonnés passerait le cap des 2000. Nous avons atteint cet objectif en décembre (deux abonnés sur trois recoivent en outre les disquettes), et nous tenons notre promesse : à partir de mars, vous aurez six numéros par an, à la fin de chaque mois impair. Le tirage augmente également, de 7000 exemplaires actuels (tous vendus) à 12000, afin de pouvoir vous offrir Pom's en kiosque. De nombreux lecteurs nous disent avoir du mal à trouver Pom's dans les boutiques de micro-informatique : si c'est le cas, n'oubliez pas de demander à votre revendeur d'en commander; sinon, cela ne s'améliorera pas ...

*Cette fois-ci, **Olivier Herz** nous a bien quitté pour le Japon, après trois mois de faux départs. Heureusement, il nous a laissé quelques contributions "posthumes" que nous vous faisons partager avec plaisir. Vous trouverez dans ce numéro son programme de "Type ahead", qui permet le stockage et la mémorisation des touches que vous enfoncez dans un "buffer" : bel utilitaire pour les stakhanovistes du clavier.*

*Nous sommes heureux d'accueillir (enfin) un journaliste non informaticien en la personne de **Daniel Garric** (la page "Futurs" du Point), auteur des "Dossiers du Futur". Daniel a découvert le Macintosh en avant-première et il vous raconte "son" Macintosh.*

*Grâce à **Gilles Chaygneaud**, la disquette de Pom's est "parlante" pour la première fois. Vous pourrez, avec son programme et un simple magnétocassette, enregistrer de la parole ou des chansons dans l'Apple qui vous les restituera ensuite à la demande. Sonorisez enfin vos programmes Basic: vous "blufferez" tout le monde. Après quelques essais et réglages, vous parviendrez à d'excellents résultats. La disquette étant déjà pleine de programmes, nous n'avons pas pu, hélas, mettre divers exemples de sonorisation.*

*Une contribution inhabituelle pour Pom's en la personne d'Aliénor, présentée par **A. Guillez** et **Y. Cherruault**. Cet article devrait intéresser particulièrement nos lecteurs scientifiques, et il démontre pour le moins que l'Apple trouve à s'employer dans les domaines les plus divers.*
*Dans un registre plus large, **Guy Lapautre** a testé pour vous le traitement de texte Gutenberg et **Alexandre Avrane** a fait de même pour Magic Window.*

***Gilles Mauffrey** vous propose un programme très simple à utiliser pour tracer des fonctions en haute-résolution et **Gérard Rigaud** présente une routine complète de tri de tableaux. **Roland Jost** vous donne pour sa part une longue liste de POKEs en tous genres. Vous pourrez également rendre l'Apple polyglotte avec le Pommesoft de **Philippe Faure**.*

*Enfin, la comparaison de programmes en assembleur ou en Applesoft vous sera grandement facilitée par les articles de **Denis Sureau** et **Gérard Michel**, qui reprend également une rubrique d'initiation à l'assembleur.*

Nous aurons l'occasion de nous rencontrer deux fois dans un avenir proche : au Spécial-Sicob (palais du CNIT - La Défense), du 14 au 19 mai, où nous occuperons le stand 630 (niveau 3, zone F), puis la semaine suivante à Micro-Expo, du 22 au 26 mai, où nous aurons le stand N44. Vous pourrez venir nous ramasser à la petite cuillère le 26 à la clôture ... A bientôt donc !

Hervé Thiriez

Ont collaboré à ce numéro : Alexandre Avrane – Gilles Chaygneaud-Dupuy – Y. Cherruault – Michel Crimont – Guy d'Herbemont – Alexandre Duback – Philippe Faure – Danier Garric – A. Guillez – Olivier Herz – Roland Jost – Guy Lapautre – Gilles Mauffrey – Gérard Michel – Gérard Rigaud – Denis Sureau – Hervé Thiriez. **Rédacteurs :** Alexandre Avrane – Gérard Michel. **Dessins :** Laurent Bidot
Directeur de la publication - rédacteur en chef : Hervé Thiriez. **Siège social :** Editions MEV – 49, rue Lamartine – 78000 Versailles – Tél. : (3) 951.24.43.
Diffusion N.M.P.P. – Sophie Mamez – Tél. : (1) 240.22.01
Composition : Télécompo – 13-15, avenue du Petit Parc – 94300 Vincennes – Tél. : 328.18.63
Impression : Rosay – 47, avenue de Paris – 94300 Vincennes – Tél. : 328.18.63

RENDEZ VOTRE APPLE * ENCORE "PLUS"

Cartes et accessoires additionnels compatibles APPLE II

FLOPPY-DRIVE POUR APPLE

NOUVEAU

3 POUCES MD3 «HITACHI»

- Capacité DD : 500 K octets.
- Nombre de pistes : 80.
- Densité : 100 TPI.
- Dimension : 90 x 40 x 150.
- Poids : 0,8 kg.

COMPLET AVEC CONTROLEUR **2950F**

Sans contrôleur ni coffret **2190 F**

disquette rigide protégée l'unité **68 F**

5 POUCES

Compatible Apple avec contrôleur

3190F

PROMOTION DISQUETTE POUR FLOPPY

5" SF-DD 48 TPI. l'unité ..... **29 F**

par 10 pièces l'unité **25 F**, par 50 pièces l'unité **19 F**

CARTE LANGAGE 16 K RAM

Pour extension du 48 K RAM en 64 K. Compatible FORTRAN PASCAL, LISP, BASIC

Entièrement équipée **695F**

CARTE D'EXTENSION 128 K RAM

Emulation disk-drive sous DOS, PASCAL ou CP/M

Entièrement équipée **2200F**

CARTE 80 COLONNES

80 car. x 24 lignes. Résolution 7 x 9. Compatible avec la plupart des traitements de texte BASIC, PASCAL, CP/M, MODEM

Entièrement équipé **895F**

CARTE Z 80

Fonctionne sous CP/M

Utilisation de tout logiciel sous CP/M.

Entièrement équipée **995F**

CARTE INTERFACE POUR 2 FLOPPY-DRIVE

Entièrement équipée **449F**

CARTE DE PROGRAMMATION 2716

895F

Programmation lecture/copie chargement de programme directement sur 2716. Entièrement équipée.

KIT GOLDEN

CARTE D'UNITE CENTRALE double processeur 6502 et Z 80. 64 K RAM

Entièrement équipée (sans ROM)

7 slots d'extensions. Fonctionne sous CP/M

CLAVIER ASC II

68 touches. Alphanumérique. Majuscules, minuscules, décimales

ALIMENTATION 220 V. 5 A

COFFRET pour carte de base et pavé numérique

KIT GOLDEN

Carte d'unité centrale avec 6502 et Z80	3380 F
Clavier ASC II	950 F
Alimentation 220 V. 5 A	799 F
Coffret	698 F
	5797 F
L'ENSEMBLE	**5199F**

Chaque élément peut-être acheté séparément.

CARTES D'INTERFAÇAGE

Carte RVB (pour moniteur couleur)	698 F
Carte «SPEETCH» en anglais	698 F
Carte musicale pour synthétiseur de son	886 F
Carte horloge	788 F
Carte vidéo MODEM	2680 F
Carte contrôleur (drive)	698 F
Carte de connection série RS 232 C	798 F

CARTE INTERFACE BUFFERISÉE IMPRIMANTE

Pour toutes marques sortie CENTRONIC'S - Buffer 64 K RAM.

Livrée équipée en 16 K (extension jusqu'à 64 K) **1750F**

TABLE GRAPHIQUE

Pour reproduction du graphisme, connectable à la place du Joy-stick **1590F**

MONITEUR COULEUR RTC en module Simple à monter

Décrit dans Radio-Plans n° 429

Tube A37 590 Y

Châssis VCC 90

Avec Péritel électronique et mécanique COMPLET

Prix **2890F**

IMPRIMANTE SEIKOSHA GRAPHIQUE COMPACTE GP 100 A

PROMO **2250F**

Interface parallèle en standard. 80 car./ligne. 50 car./sec. Impression en simple ou double largeur Papier normal. Entraînement par tracteurs ajustables.

INTERFACES POUR GP100 A

APPLE II ou IIE avec câble	990 F
Série RS 232	795 F
ZX 81	850 F

Câbles pour SANYO 280F, HC 25 280F, ORIC 280F, TO 7 280F

Papier pour GP 100 Les 1000 feuilles	160 F
Ruban encreur GP 100	99 F

NOUVEAU

IMPRIMANTE GP 50A SEIKOSHA

- Entraînement à friction
- Graphique
- 2 épaisseurs de caractères
- Interface parallèle compatible CENTRONICS

1250F

JOY-STICK

PROMO **219F**

équipé de 2 trimes pour recherche du point zéro

PROMO **169F**

MONITEURS

ZENITH 12" écran vert **999F**

OCEANIC 12" couleur **3500F**

VENTILATEUR «FAN» 495 F

EFFACEUR D'EPROM EN KIT

Complet avec notice **180F**

* APPLE est une marque déposée et appartient à APPLE COMPUTER S.A.

CONDITIONS GENERALES DE VENTES PAR CORRESPONDANCE
Pour éviter les frais de contre-remboursement, nous vous conseillons de régler vos commandes intégralement (y compris frais de port). FORFAIT DE PORT : 25 F.

ACER MICRO

42, rue de Chabrol, 75010 Paris.
Tél. 770.28.31.

Le Bavard

Gilles Chaygneaud-Dupuy

Saviez-vous que votre Apple peut dire quelques mots, sans aucune prothèse, pour peu que vous les lui appreniez !
Cela est possible grâce au système "Le Bavard", qui se compose d'un petit module en langage machine et d'un programme en Applesoft servant d'éditeur.

Le module PARLE

Le programme source a été écrit avec l'assembleur Dos Tool Kit.
Ce module permet de mettre sous forme numérique des mots préalablement enregistrés sur un magnétophone et de les stocker en mémoire. Une deuxième sous-routine (voir le programme d'édition) permet de restituer les sons à différentes vitesses, soit sur le haut-parleur de l'Apple, soit sur un amplificateur extérieur.

La restitution n'est pas toujours très intelligible sur l'Apple, car ce dernier ne dispose que d'une seule voix pour sortir l'ensemble des fréquences simultanées.

Le programme "BAVARD"

Il vous propose plusieurs fonctions :

1 - Enregistrer un texte en mémoire à partir d'un magnétophone connecté à l'Apple. Le magnétophone étant sur écoute, il suffit de taper une touche quand arrive le texte voulu, puis d'arrêter le stockage par la frappe d'une autre touche.

2 - Ecouter le texte actuellement en mémoire.

3 - Stocker sur disquette le texte qui se trouve en mémoire sous forme d'un fichier binaire dont vous devez préciser le nom.

4 - Relire en mémoire un texte préalablement stocké sur disquette.

5 - Choisir l'organe de sortie qui peut être le haut parleur de l'Apple ou un organe extérieur, en passant par la sortie magnétophone.

6 - Editer le catalogue de la disquette pour sélectionner les fichiers à relire.

7 - Changer la vitesse de restitution des sons, qui peut aller de 0 à 9, 0 donnant la voix la plus aiguë et 9 la plus grave.

8 - Visualisation des sons enregistrés en mémoire sous forme de courbes de fréquence, avec en abscisse l'adresse de l'octet et en ordonnée la valeur de la fréquence vocale que contient cet octet. Les silences correspondent à des zéros en haut de l'écran et l'on va de l'aigu au grave en allant du haut vers le bas de l'écran.
Il faut donner en paramètre l'adresse de départ de cette visualisation, 0 étant le début de l'enregistrement.

Utilisation dans vos programmes

Si vous voulez sonoriser vos programmes Applesoft au moyen de textes enregistrés par "Le Bavard", il faut charger par BLOAD le programme PARLE1.OBJ et le fichier contenant le texte concerné. Ce dernier peut être implanté n'importe où en mémoire, mais il faut en préciser l'adresse de départ par un POKE aux adresses 0 et 1 avant d'appeler le module par CALL 819. Par exemple :

```
PRINT CHR$(4) "BLOAD PARLE1.OBJ"
PRINT CHR$(4)"BLOAD COCORICO.A$4000"
POKE 0,0 : POKE 1,64
CALL 819
```

Pour faire varier la vitesse, il faut effectuer un "POKE 846, 252–V" avant l'appel du module, V étant un chiffre compris entre 0 et 9.

Pour gérer une table de mots à prononcer (afin de sonoriser le clavier, par exemple), il faut les séparer les uns des autres dans leur représentation numérique par un octet contenant $FF, qui marque la fin de l'enregistrement. Indiquez ensuite l'adresse de début du mot voulu avant chaque appel au module PARLE1.OBJ. ■

Programme BAVARD

```
0  REM   PROGRAMME "LE BAVARD"
                   (APPLESOFT)
1  PRINT "BLOADPARLE1.OBJ"
2  PRINT "BLOADCOCORICO"
3  FOR I = 2 TO 8 STEP 2: POKE 846
     ,254 - I: POKE 0,0: POKE 1,64
     : CALL 819: NEXT
4  POKE 846,254 - 6
6  PRINT "BLOADBAVARD.MESS"
10  TEXT : HOME
15  INVERSE : VTAB 2: PRINT ":"; SPC(
     12): HTAB 29: PRINT  SPC( 12)
     ;":": VTAB 24: PRINT ":"; SPC(
     37);":";
20  FOR I = 3 TO 23: VTAB I: HTAB
     1: PRINT " ";: HTAB 40: PRINT
     " ";: NEXT
22  VTAB 1: HTAB 15: PRINT ":
            :": HTAB 15: PRINT ": L
     E BAVARD :":: HTAB 15: PRINT
     ":            :"
30  NORMAL
35  POKE  - 16368,0
40  HTAB 6: VTAB 6: PRINT "1-ENREG
     ISTRER UN TEXTE"
50  HTAB 6: VTAB 8: PRINT "2-ECOUT
     ER UN TEXTE"
60  HTAB 6: VTAB 10: PRINT "3-SAUV
     ER UN TEXTE SUR DISQUE"
70  HTAB 6: VTAB 12: PRINT "4-CHAR
     GER UN TEXTE EN MEMOIRE"
80  HTAB 6: VTAB 14: PRINT "5-SELE
     CTIONER LA SORTIE"
```

```
90  HTAB 6: VTAB 16: PRINT "6-AFFI
     CHER LE CATALOGUE"
92  HTAB 6: VTAB 18: PRINT "7-CHAN
     GER LA VITESSE D'ECOUTE"
95  HTAB 6: VTAB 20: PRINT "8-VISU
     ALISER UN TEXTE "
97  HTAB 6: VTAB 22: PRINT "9-SORT
     IR DU PROGRAMME"
99  IF N = 0 GOTO 2000
100  HTAB 35: VTAB 22: GET N
110  ON N GOTO 1000,2000,3000,4000
     ,5000,6000,7000,8000,9000
1000  HOME
1010  VTAB 10: PRINT "VOUS METTEZ
     VOTRE MAGNETOPHONE EN ROUTE E
     T VOUS APPUYEZ SUR RETURN POU
     R          DEBUTER L'ENREGISTR
     EMENT"
1015  PRINT "PUIS APPUYEZ DE NOUVE
     AU SUR RETURN       POUR L'ARR
     ETER."
1020  INPUT A$
1025  POKE 00,00: POKE 01,64
1030  CALL 768
1040 FIN =  PEEK (01) * 256 +  PEEK
     (00)
1050 LG = FIN - 16384 + 10
1990  GOTO 10
2000  POKE 00,00: POKE 01,64
2010  CALL B19
2020 N = 1: GOTO 10
3000  HOME
3010  VTAB 10: INPUT "NOM DU FICHI
     ER A SAUVER ? ";A$
3020  PRINT "BSAVE";A$;",A$4000,L"
     ;LG
3990  GOTO 10
4000  HOME
4005  VTAB 10: INPUT "NOM DU TEXTE
      A CHARGER ? ";A$
4010  PRINT "BLOAD";A$;",A$4000"
4020 LG =  PEEK (43616) +  PEEK (4
     3617) * 256
4990  GOTO 10
5000  HOME
5010  VTAB 10: PRINT "VOULEZ VOUS
     ECOUTEZ VOTRE TEXTE :"
5020  PRINT "  A-SUR L'APPLE"
5030  PRINT "  B-SUR UN AMPLI EXTE
     RIEUR"
5040  PRINT : INPUT "VOTRE CHOIX ?
      ";A$
5050  IF A$ = "A" THEN  POKE 851,4
     8: GOTO 10
5060  IF A$ = "B" THEN  POKE 851,3
     2: GOTO 10
5070  GOTO 5000
5990  GOTO 10
6000  PRINT : PRINT  CHR$ (4)"CATA
     LOG"
6015  INPUT A$
6020  GOTO 10
7000  HOME
7005 V = 252 -  PEEK (846)
7010  VTAB 10: PRINT "LA VITESSE E
     ST ACTUELLEMENT DE ";V
7020  PRINT : PRINT "    A COMBIEN
      LA VOULEZ VOUS ? ";: GET V
7030  POKE 846,252 - V
7040  GOTO 2000
8000  HOME
8002  VTAB 22: INPUT "ADRESSE DEBU
     T A VISUALISER ";A$
8004  IF A$ = "" THEN  TEXT : GOTO
     10
8005  HGR : HCOLOR= 3
8006 A =  VAL (A$)
8010  FOR I = 0 TO 259
8020 Y =  PEEK (16384 + A + I)
8025 Y = Y / 255 * 160
8030  HPLOT I,Y
8040  NEXT
8050  GOTO 8000
9000  END
```

Programme PARLE1.S
DOS TOOL KIT

```
0300:          1        ORG  $300      PROGRAMME PARLE
C030:          2 BRUIT  EQU  $C030
C060:          3 CASSET EQU  $C060
0000:          4 ADD    EQU  0
C000:          5 CLAVIER EQU $C000
0006:          6 SON    EQU  $6
0007:          7 FREQ   EQU  $7
0300:AD 00     8 DEBUT  LDY  #0
0302:B4 07     9        STY  FREQ
0304:E6 07    10        INC  FREQ
0306:AD 60 C0 11        LDA  CASSET
0309:85 06    12        STA  SON
030B:AD 60 C0 13 INF    LDA  CASSET
030E:45 06    14        EOR  SON
0310:30 04    15        BMI  SUIVAN
0312:E6 07    16        INC  FREQ
0314:D0 F5    17        BNE  INF
0316:A5 07    18 SUIVAN LDA  FREQ
0318:C9 FF    19        CMP  #$FF
031A:D0 02    20        BNE  *+4
031C:A9 FE    21        LDA  #$FE
031E:91 00    22        STA  (ADD),Y
0320:E6 00    23        INC  ADD
0322:D0 02    24        BNE  ADD2
0324:E6 01    25        INC  ADD+1
0326:2C 00 C0 26 ADD2   BIT  CLAVIER
0329:30 03    27        BMI  FIN
032B:4C 00 03 28        JMP  DEBUT
032E:A9 FF    29 FIN    LDA  #$FF
0330:91 00    30        STA  (ADD),Y
0332:60       31        RTS
0333:B1 00    32 DIT    LDA  (ADD),Y
0335:85 06    33        STA  SON
0337:C9 FF    34        CMP  #$FF
0339:F0 F7    35        BEQ  FIN+4
033B:2C 00 C0 36        BIT  CLAVIER
033E:30 F2    37        BMI  FIN+4
0340:48       38        PHA
```

```
0341:EA        39 BOUCLE2 NOP
0342:EA        40         NOP
0343:EA        41         NOP
0344:EA        42         NOP
0345:EA        43         NOP
0346:EA        44         NOP
0347:EA        45         NOP
0348:EA        46         NOP
0349:EA        47         NOP
034A:EA        48         NOP
034B:C6 06     49         DEC  SQN
034D:D0 F2     50         BNE  BOUCLE2
034F:68        51         PLA
0350:F0 03     52         BEQ  SAUT
0355:E6 00     54 SAUT    INC  ADD
0357:00 DA     55         BNE  DIT
0359:E6 01     56         INC  ADD+1
0358:4C 33 03  57         JMP  DIT
0352:A0 30 C0  53         LDA  BRUIT
```

Récapitulation

```
0300- A0 00 B4 07 E6 07 A0 60
0308- C0 85 06 AD 60 C0 45 06
0310- 30 04 E6 07 D0 F5 A5 07
0318- C9 FF D0 02 A9 FE 91 00
0320- E6 00 D0 02 E6 01 2C 00
0328- C0 30 03 4C 00 03 A9 FF
0330- 91 00 60 B1 00 85 06 C9
0338- FF F0 F7 2C 00 C0 30 F2
0340- 48 EA EA EA EA EA EA EA
0348- EA EA EA C6 06 D0 F8 68
0350- F0 03 AD 30 C0 E6 00 D0
0358- DA E6 01 4C 33 03
```

Une astuce pour supprimer les REMs

Guy d'Herbemont

S'il est vrai que les REMs ralentissent et encombrent un programme, il ne l'est pas moins qu'elles contribuent souvent à sa bonne compréhension pour quiconque ne l'a pas écrit.
La solution consiste donc à conserver deux versions d'un même programme, avec et sans commentaires, ce qui justifie le besoin d'une procédure automatique de suppression des REMs à l'intérieur d'un programme en Applesoft.

Celle que je vous propose repose sur un principe très simple : si l'on remplace tous les codes REM (178) par 00, le programme n'est plus exécutable mais il se liste normalement sans les REMs (le 00 étant interprété comme une fin de ligne). Il suffit donc de stocker cette liste dans un fichier texte NEW-LIST, et de faire "NEW" puis "EXEC NEW-LIST", pour récupérer en mémoire une version exécutable du programme, sans REMs, qu'il ne reste qu'à sauvegarder.

Le programme FAIT SANS-REM listé ci-après construit un fichier SANS-REM qui réalise les opérations suivantes :

- Chargement en mémoire des lignes 60005 à 60096. Ce petit programme assure la transformation des codes REM et la sauvegarde de la liste en résultant dans le fichier NEW-LIST.
- Exécution de l'ordre "RUN 60005" pour effectuer le traitement ci-dessus.
- Exécution de "NEW" puis "EXEC NEW-LIST" pour récupérer en mémoire le programme original modifié.

Une fois ce fichier construit, l'utilisation de la procédure se réduit à :

- LOAD "Programme à modifier"
- EXEC SANS-REM

Un LIST à l'issue des opérations vous permettra de vérifier que les REMs ont bien disparu (les ":" qui précèdent éventuellement les instructions REM sont en revanche conservés), et il ne vous reste plus qu'à sauver la version allégée de votre programme.

A noter que les lignes 60005 à 60096 ne doivent impérativement contenir aucune REM et que, bien sûr, le programme original ne doit comporter aucune ligne de numéro supérieur à 59999. ■

Programme FAIT SANS-REM

```
1  REM  PROGRAMME POUR CONSTITUER
2  REM  LE FICHIER EXEC
3  REM  "SANS-REM"
4  REM  ==========================

10 D$ =  CHR$ (13) +  CHR$ (4):N$ =
     "SANS-REM": PRINT D$"OPEN"N$D
     $"DELETE"N$D$"OPEN"N$D$"WRITE
     "N$
20  POKE 33,30: LIST 60005,: PRINT
     "RUN 60005": PRINT "NEW": PRINT
     "EXEC NEW-LIST": PRINT D$"CLO
     SE": TEXT : END
60005 DEB = 2049
60010 X = DEB + 3
60015 X = X + 1:BYT =  PEEK (X)
60020  IF BYT = 0 THEN  GOSUB 6006
     0: ON DEB = 0 GOTO 60070: ON
     DEB <  > 0 GOTO 60010
60027  IF BYT <  > 178 GOTO 60015
60030  POKE X,0
60040  GOSUB 60060: ON DEB = 0 GOTO
     60070
60050  GOTO 60010
60060 DEB =  PEEK (DEB) + 256 *  PEEK
     (DEB + 1): RETURN
60070  POKE 33,30
60072 D$ =  CHR$ (13) +  CHR$ (4):
     N$ = "NEW-LIST": PRINT D$"OPE
     N"N$D$"DELETE"N$D$"OPEN"N$D$"
     WRITE"N$
60080  LIST ,59999
60090  PRINT D$"CLOSE"
60095  TEXT : HOME
60096  END
```

Apprentissage de l'assembleur

Gérard Michel

Cet article constitue la reprise d'une série abordée dans les premiers numéros de Pom's. Si l'on en juge par votre courrier, vous étiez nombreux à souhaiter le retour d'une rubrique consacrée à l'apprentissage de l'assembleur; nous allons donc nous efforcer de vous donner satisfaction, conformément aux objectifs de cette revue.

Rappelons tout d'abord, pour éviter toute confusion, que l'assembleur est déjà un langage "évolué", à ne pas confondre avec le langage machine. En matière de logiciel (et en simplifiant), la seule chose que votre Apple puisse comprendre sans l'aide du moindre compilateur ou interpréteur, c'est de l'hexadécimal, autrement dit des codes comme ceux que vous trouvez dans Pom's à la suite d'un programme en assembleur sous le titre "récapitulation" (20 C1 FB ... par exemple).

Un assembleur est donc un langage constitué d'instructions plus facilement compréhensibles que l'hexadécimal, théoriquement plus faciles également à mémoriser (pour les anglicistes du moins), doublé d'un "traducteur" qui transforme automatiquement ces instructions en code exécutable par la machine (cette opération est baptisée "assemblage"). C'est pourquoi l'on parle toujours de programme source (en langage assembleur) et de programme objet (en hexadécimal), résultat de la traduction du premier.
Cela signifie en particulier que vous ne pouvez pas programmer en assembleur si vous ne possédez pas un logiciel "éditeur/assembleur" tel que LISA, BIG MAC, DOS TOOL KIT ou autre, autant de noms qui doivent vous être maintenant familiers si vous êtes un fidèle lecteur de Pom's. Par contre, vous pouvez toujours écrire un programme par le biais du code hexadécimal, du moins dans un premier temps, ne serait ce que pour vous faire une idée de ce que pourrait vous apporter l'assembleur. C'est pourquoi nous vous donnerons dans cet article le code correspondant à chacune des instructions présentées; de plus, dans toute la mesure du possible, nous vous fournirons une version des opérations réalisées sous forme d'instructions en Applesoft, ce qui devrait aider à l'analyse et à la compréhension des mécanismes mis en oeuvre, que nous allons aborder sans plus attendre.

Variables et registres

Comme tous les langages, l'assembleur permet de travailler sur et avec des variables. En première approximation, une variable est une case mémoire contenant un octet (deux chiffres hexadécimaux), définie par son adresse. Ainsi, $6 (soit 6 en décimal) ou $18 (soit 24 en décimal), par exemple, peuvent constituer l'adresse d'une variable dans laquelle le programme ira lire ou stocker des valeurs. Rappelons que, par convention, un nombre précédé de $ est toujours un nombre hexadécimal.
Il est d'ailleurs important de distinguer dès maintenant la variable (ou l'adresse) et son contenu. Lorsque l'on écrit simplement $18, on fait référence à la case mémoire dont l'adresse hexadécimale est $18; si l'on écrit #$18 par contre, on désigne la valeur $18 (ou 24 en décimal). De même, chaque fois qu'une instruction de l'assembleur porte sur la manipulation d'une variable, c'est son contenu qui s'en trouve affecté (en BASIC, A=B transfère également le contenu de B dans A mais ne remplace pas le A par un B).
Notons en outre une différence de régime, sur laquelle nous reviendrons plus tard, entre les adresses dites "page zéro" et les autres. Une adresse est en page zéro si elle est inférieure à 256 (de $0 à $FF en hexadécimal) et peut donc être représentée par un seul octet.

Affectation d'une valeur

En mode immédiat, l'affectation d'une valeur à une variable peut se faire directement. Supposons par exemple que l'on veuille mettre #$FF (soit 255 en décimal) à l'adresse $18, en partant du BASIC, le processus serait le suivant :

CALL −151 (pour accéder au moniteur)
*18: FF "RETURN" (l'étoile est le "prompt" du moniteur et est affichée par le système, le reste doit être tapé au clavier, sans oublier les ":"). "RETURN" signifie tout simplement "Appuyer sur la touche RETURN".

Ceci est équivalent à l'instruction Applesoft : POKE 24,255 "RETURN".
Pour lire le contenu de l'adresse $18 à partir du moniteur, tapez :
18 "RETURN"
et la machine répondra :
*0018-FF
Vous auriez pu obtenir le même résultat (255 en l'occurrence) avec les instructions Applesoft : PRINT PEEK(24).

Les opérations de lecture/écriture du contenu d'une variable à partir d'un programme en assembleur supposent l'emploi des registres du microprocesseur car, si l'on peut faire "A=255" en BASIC, on ne peut pas faire directement "$18=#$FF" en assembleur (la définition directe de constantes est toutefois possible, mais ce n'est pas notre propos pour l'instant). Un registre peut se définir comme une variable particulière, dont le nom reste toujours le même, que le microprocesseur utilise pour communiquer avec la mémoire de l'ordinateur. La "case mémoire" correspondant à un registre se trouve, en quelque sorte, dans le microprocesseur et le programmeur n'a jamais besoin de la définir lui-même.
Dans ce domaine, trois registres peuvent nous être utiles : l'accumulateur (désigné plus simplement par A), le registre d'index X et le registre d'index Y. Chacun de ces registres peut contenir un octet et prendre ainsi une valeur comprise entre 0 et 255 ($0 et $FF).
Si nous nous acharnons toujours à mettre #$FF dans la case $18, trois séquences d'instructions sont ainsi disponibles :

1) LDA #$FF (A9 FF)
   STA $18 (85 18)

ce qui signifie "charger accumulateur (LoaD A) avec la valeur #$FF" puis "stocker la valeur de A (STore A) dans la case d'adresse $18". En Applesoft, cela pourrait s'écrire : A=255:POKE 24,A
Les codes entre parenthèses ne sont pas de l'assembleur mais correspondent à ce qu'il faudrait taper au clavier pour rentrer ces instructions directement en hexadécimal. Ainsi, A9 est le code de LDA lorsque celui-ci est suivi d'une valeur (FF) et 85 est le code de STA lorsque celui-ci est suivi d'une adresse en page zéro (18). Si vous voulez par exemple implanter ces instructions à partir de l'adresse $300 (768 en décimal), il vous faudra taper, à partir du moniteur (derrière l'étoile de "prompt") :
300: A9 FF 85 18 "RETURN"
Si vous tapez ensuite :
300L "RETURN"
le système vous répondra :
300-A9 FF LDA #$FF
302-85 18 STA $18
plus d'autres choses sans signification pour nous et qui dépendent de ce qui se trouve dans votre Apple à par-

tir de l'adresse $304 au moment où vous exécutez ce test.
La commande "L" (pour LIST), tapée derrière une adresse, demande en effet au moniteur d'essayer de désassembler les codes hexadécimaux stockés à partir de cette adresse pour en faire 20 lignes d'instructions en langage assembleur. Bien sûr, le résultat en est plus ou moins cohérent selon les codes rencontrés, mais n'oubliez pas cette aide que peut vous apporter le système car elle s'avère toujours utile pour le déchiffrage de routines dont on ne possède pas le programme source.

2) LDX #$FF (A2 FF)
STX $18 (86 18)

Compte tenu de ce qui précède, la signification de ces deux instructions ne doit plus vous échapper (on passe par le registre X au lieu de passer par l'accumulateur).

3) LDY #$FF (A0 FF)
STY $18 (84 18)

Même chose avec le registre Y.

Utilisation d'étiquettes

Les assembleurs "symboliques" (toujours LISA, BIG MAC ...), permettent de donner un nom, ou une étiquette, aux variables. Cela rend le programme source plus parlant et plus facile à analyser. LIGNE, par exemple, est d'une interprétation plus directe que $C6. En outre, l'affectation d'une étiquette simplifie considérablement la mise à jour d'un programme source si l'on se trouve contraint, pour une raison ou pour une autre, de modifier l'adresse d'une variable. Il suffit alors de changer la définition de l'étiquette, alors qu'il faudrait autrement rechercher tous les endroits où apparaît l'adresse concernée pour la modifier.
D'un assembleur à l'autre, la définition de ces étiquettes revêt une syntaxe légèrement différente. Pour notre chère adresse $18, on pourrait ainsi avoir :
ADR EPZ $18 (si on veut l'appeler ADR), ou
ADR EQU $18, ou
ADR = $18 .
C'est à la documentation de votre assembleur qu'il faut vous reporter pour connaître la bonne syntaxe. Notons simplement qu'une étiquette doit toujours être définie quelque part dans le programme source pour être utilisée dans une instruction sans provoquer d'erreur à l'assemblage (la solution la plus claire consiste à les mettre toutes en tête de programme, comme le font les auteurs des articles publiés dans Pom's).

Pour en revenir à notre exemple, si l'on suppose que l'on a baptisé la case $18 du nom ADR, on pourrait écrire en assembleur :
LDA #$FF (A9 FF)
STA ADR (85 18)
Notez que le nom symbolique n'est valable que pour le programme source; le code hexadécimal reste toujours le même et fait référence à $18.
En Applesoft, cela reviendrait à :
ADR=24:A=255:POKE ADR,A

Lecture d'une case mémoire

Pour lire le contenu d'une case-mémoire, les trois registres peuvent également être utilisés, d'où trois solutions disponibles :

1) LDA $18 ou LDA ADR (A5 18)
2) LDX $18 ou LDX ADR (A6 18)
3) LDY $18 ou LDY ADR (A4 18)

Ces trois instructions reviennent à charger respectivement l'Accumulateur, le registre X ou le registre Y, avec le contenu de l'adresse $18.

Vous aurez sans doute remarqué que les codes des instructions LDA, LDX et LDY ne sont pas les mêmes que lorsqu'on leur donnait directement la valeur #$FF. Chaque type d'opération possède un code particulier: ainsi :

1) Pour LDA :

- LDA valeur = A9 valeur (exemple : LDA #$30 = A9 30)
- LDA contenu d'une adresse en page zéro = A5 adresse (exemple : LDA $24 = A5 24)
- LDA contenu d'une adresse hors de la page zéro = AD octet bas de l'adresse/octet haut de l'adresse.

Lorsqu'une adresse s'écrit sur deux octets, soit 4 chiffres hexadécimaux, l'octet bas correspond aux deux chiffres de droite et l'octet haut aux deux chiffres de gauche. Par exemple :
LDA $300 = AD 00 03
LDA $FDED = AD ED FD
C'est une sale manie, mais l'Apple prend toujours les adresses à l'envers de cette façon !

2) Pour LDX

- LDX valeur = A2 valeur
- LDX contenu d'une adresse page zéro = A6 adresse
- LDX contenu d'une adresse quelconque = AE octet bas/octet haut (exemple : LDX $40F = AE 0F 04)

3) Pour LDY

- A0 valeur
- A4 adresse
- AC octet bas/octet haut

Opérations sur les registres

Nous ne présenterons pour l'instant que celles qui nous seront utiles pour la suite de cet article.

1) INX (code E8) : incrémente X de 1 (équivalent à X=X+1).
2) DEX (code CA) : décrémente X de 1 (équivalent à X=X−1).
3) INY (code C8) : incrémente Y de 1.
4) DEY (code 88) : décrémente Y de 1.
5) TAX (code AA) : transfère le contenu de A dans X, sans modifier celui de A (équivalent à X=A).
6) TAY (code A8) : transfère le contenu de A dans Y (Y=A).
7) TXA (code 8A) : transfère le contenu de X dans A (A=X).
8) TYA (code 98) : transfère le contenu de Y dans A (A=Y).

Egal à zéro ou différent de zéro ?

Parmi les instructions de test et de branchement disponibles en assembleur, deux nous seront également utiles dans un futur immédiat :

1) BEQ (code F0)

En Applesoft, l'instruction BEQ serait équivalente à IF "dernier résultat"=0 THEN branchement.
Si le dernier résultat n'est pas nul, on passe simplement à l'instruction suivante.
Compte tenu des instructions que nous avons déjà examinées, nous pouvons dire à ce stade que le "dernier résultat" en question correspond à la valeur du dernier registre manipulé. Par exemple :

- Après LDA $18, BEQ provoquera un branchement si $18, et donc l'accumulateur, contient 0.
- Après DEY, BEQ provoquera un branchement si Y tombe à 0.
- Après TXA, BEQ provoquera un branchement si X, et donc A, contient 0

L'endroit vers lequel s'effectue le branchement ne pose pas de problème si l'on écrit un programme source en assembleur symbolique. Comme pour les variables, il est en effet possible de mettre des étiquettes en regard des instructions: le branchement se fait alors par référence à cette étiquette (voir plus bas). Si l'on travaille directement au niveau de l'hexadécimal, il faut alors savoir que BEQ ne fait pas de branchement sur une adresse, mais en fonction d'un déplacement vers le haut ou vers le bas. La valeur de ce déplacement doit suivre le code F0. Voici quelques exemples :

- F0 00 : le branchement se fait à l'octet qui suit 00 (si le F0 se trouve à l'adresse $300, BEQ branchera à l'adresse $302 si le dernier résultat est nul). Une telle instruction serait donc inutile puisque l'on arriverait à la même adresse quel que soit le résultat du test.

- F0 01 : branchement à "adresse de F0 + 3"
- F0 02 : branchement à "adresse de F0 + 4"
- F0 FF : branchement à "adresse de F0 + 1" (ce qui provoquera une erreur puisque l'on retombe sur FF, qui n'est pas le code d'une instruction). La valeur $FF correspond en fait à −1 en décimal : lorsqu'il "lit" F0, le microprocesseur "pointe" aussitôt vers l'octet suivant (FF en l'occurrence), de même lorsqu'il lit le FF qui recule le pointeur de 1 et le ramène donc sur FF.
- F0 FE : branchement sur l'adresse de F0 (boucle infinie !)
- F0 FD : branchement à "adresse de F0 − 1"
- F0 FC : branchement à "adresse de F0 − 2"

De manière générale, si A est l'adresse à laquelle se trouve le code F0, un déplacement compris entre $0 et $7F branchera sur une adresse comprise entre A+$2 et A+$81 (+2 à +129 en décimal), tandis qu'un déplacement compris entre $FD et $80 branchera sur une adresse comprise entre A−$1 et A−$7C (−1 à −124 en décimal). Cela signifie en particulier que, à partir d'une adresse A, on ne peut effectuer de saut "en avant" que jusqu'à A+129, et de saut "en arrière" que jusqu'à A−124.

Il est clair que tout ceci n'est pas simple à manipuler, et l'assembleur symbolique est vraiment d'un grand secours pour tout ce qui concerne les branchements, car il calcule le déplacement à votre place.

2) BNE (code D0)

En Applesoft, BNE pourrait se traduire par IF "dernier résultat" <>0 THEN branchement (vous deviez vous en douter un peu...).
Les principes sont identiques à ceux qui régissent l'instruction BEQ en ce qui concerne le "dernier résultat" et les branchements.

Les modes d'adressage

Adressage direct indexé

Nous avons déjà rencontré la technique d'adressage la plus simple, que l'on peut baptiser "adressage absolu", qui consiste à donner directement l'adresse de chaque variable (ainsi dans STA $18 ou LDA $06).

Supposons maintenant que l'on veuille mettre la valeur 0 dans les adresses $8001 à $800A. Une solution consisterait à faire successivement référence aux adresses $8001, $8002, $8003... Ce n'est certainement pas la plus économique en place-mémoire, ni en effort musculaire au clavier.

Il est heureusement possible d'utiliser les registres X et Y pour effectuer ce travail à moindre coût (ce qui vous expliquera par la même occasion leur nom de registre "index"). Les instructions suivantes réalisent l'opération :

```
LDA #0 (A9 00)
LDX #$0A (A2 0A)
STOCKE STA $8000,X (9D 00 80)
DEX (CA)
BNE STOCKE (D0 FA)
```

En Applesoft, cela pourrait s'écrire :

```
10 A=0
20 X=10
30 ST=32768 (soit $8000)
40 POKE ST+X,A
50 X=X−1
60 IF X<>0 THEN 40
```

La valeur de X est ajoutée à l'adresse de base ($8000, donnée ici en clair, mais qui peut être également une adresse symbolique comme ADR avec STA ADR,X) pour donner l'adresse définitive où est stockée la valeur 0. X passe ainsi de 10 à 0 (instruction DEX); tant qu'il est différent de 0, on "remonte" à l'instruction de stockage à l'adresse calculée (indexée); lorsqu'il est nul, le branchement ne se fait plus et on passe alors à l'instruction suivant BNE STOCKE.

Le même résultat serait obtenu avec le registre Y :

```
LDA #0 (A9 00)
LDY #$0A (A0 0A)
STOCKE STA $8000,Y (99 00 80)
DEY (88)
BNE STOCKE (D0 FA)
```

L'instruction LDA, comme STA, peut utiliser l'adressage indexé par X ou Y. Les différents codes hexadécimaux sont listés ci-dessous :

1) Pour LDA (ADS désigne une adresse symbolique) :
- LDA adresse quelconque,X = BD octet bas/octet haut = LDA $aaaa,X ou LDA ADS,X
- LDA adresse quelconque,Y = B9 octet bas/octet haut = LDA $aaaa,Y ou LDA ADS,Y
- LDA adresse page zéro,X = B5 adresse = LDA $aa,X ou LDA ADS,X
- Pas d'adressage indexé par Y en page zéro.

2) Pour STA :
- STA adresse,X = 9D octet bas/octet haut = STA $aaaa,X ou STA ADS,X
- STA adresse,Y = 99 octet bas/octet haut = STA $aaaa,Y ou STA ADS,Y
- STA adresse page zéro,X = 95 adresse = STA $aa,X ou STA ADS,X

Chacun des registres permet d'adresser une zone de mémoire de 256 octets, d'une adresse de base $A jusqu'à $A+$FF. Il faut bien se rappeler que, quel que soit le type d'indexation, l'adressage ne peut porter que sur 256 octets à partir d'une même adresse de base; si l'on doit travailler sur une zone plus grande, il devient nécessaire de manipuler également cette adresse. De manière générale, l'adresse variera selon un pas de 256 octets, par exemple : $8000 (on peut aller de $8000 à $80FF avec un index), $8100 (de $8100 à $81FF avec un index)... Vous aurez certainement noté que cette variation se traduit par une incrémentation de 1 en 1 de l'octet haut de l'adresse.

Adressage indirect indexé

Il nous reste à voir une technique d'adressage plus complexe mais qui n'est pas moins utilisée pour autant, au contraire. Elle fait appel à l'indexation par X ou Y, mais elle est dite également "indirecte" car l'adresse définitive est calculée à partir de la valeur contenue dans l'adresse de base et non à partir de l'adresse de base elle-même. Cette dernière (baptisons la AD) doit obligatoirement se trouver en page zéro et le système prendra l'adresse contenue selon le principe : valeur contenue dans AD=octet bas de l'adresse et valeur contenue dans AD+1=octet haut de l'adresse.

Adressage pré-indexé par X

Le mécanisme en est le suivant :
- soit AD l'adresse de base en page 0,
- le système lui ajoute la valeur de X,
- le contenu de AD+X et AD+X+1 donne une adresse A0 sur laquelle portera le travail effectif de lecture/écriture.

En Applesoft, cela reviendrait à une suite d'instructions du type : AD=6 : X=2 : AX=AD+X : A0 = PEEK(AX)+256*PEEK(AX+1) : POKE A0,0 (si l'on veut, par exemple, mettre 0 dans A0).
Du fait que les adresses contenues comportent toujours deux octets, même si elles sont inférieures à 256 (l'octet de poids fort vaut alors 0), X ne doit prendre que des valeurs multiples de 2 pour que l'adressage donne un résultat cohérent (0 pour atteindre la première adresse, 2 pour la seconde...).

Cette technique d'adressage permet de gérer des tables de "pointeurs" stockées en page zéro à partir d'une adresse donnée (chaque adresse contenue pointe vers une adresse définitive à manipuler par le programme). Elle autorise également l'usage d'adresses calculées par le programme (ou par l'Apple lui-même) et que vous ne connaissez

pas encore au moment où vous l'écrivez : il vous suffit de choisir une adresse de base en page zéro, à partir de laquelle seront stockés les résultats des calculs, et d'écrire votre programme en faisant uniquement référence à cette adresse.
Les instructions LDA et STA peuvent faire appel à cette technique, avec la syntaxe suivante :

- LDA (ADR.X) = A1 adresse page zéro = LDA ($aa.X)
- STA (ADR.X) = 81 adresse page zéro = STA ($aa.X)

Adressage post-indexé par Y

Les principes sont identiques en ce qui concerne l'adresse base en page zéro et l'adresse contenue, mais le fonctionnement de l'indexation est radicalement différent :

- soit AD l'adresse de base,
- elle contient une adresse A0,
- le systéme ajoute la valeur de Y à A0 pour obtenir l'adresse définitive.

En Applesoft : AD=6 : A0=PEEK(AD)+256*PEEK(AD+1) : Y=1 : POKE A0+Y,0 (par exemple).
L'indexation s'applique donc à l'adresse contenue et non plus à l'adresse de base.

Ce mode d'adressage est très utilisé par le système lui-même et, par suite, il le sera souvent aussi par le programmeur. Le but principal est toujours de pouvoir écrire un programme en gérant des adresses que l'on ne connaît pas encore ou qui peuvent se modifier, en cours d'exécution ou selon le contexte. L'adresse contenue doit fournir le début d'une zone de mémoire sur laquelle on veut travailler (chaque variable de la zone pouvant être touchée au moyen de l'index Y). Sur l'Apple, ce type d'adresse est particulièrement répandu : adresse de la fin d'un programme BASIC, adresse d'une ligne d'écran texte ou graphique, adresse de la zone des chaînes de caractères, adresse d'une ligne de programme BASIC...

Le "GOSUB" de l'assembleur

Quelques mots seulement, car le principe est très proche de celui utilisé en Applesoft. L'équivalent de GOSUB est l'instruction JSR (code 20). En assembleur, on peut faire JSR $FBC1 (par exemple) ou JSR ETIQUETTE (sous réserve bien sûr que ETIQUETTE soit défini quelque part dans le programme). En hexadécimal, l'adresse où commence la sous-routine doit suivre le code 20 dans l'ordre habituel octet bas/octet haut (20 C1 FB pour JSR $FBC1).
L'instruction RTS (code 60), quant à elle, joue exactement le même rôle que RETURN.

Récapitulation illustrée

Le petit programme ci-dessous n'emploie que des instructions examinées plus haut, à l'exception de ORG. En fait, il ne s'agit pas vraiment d'une instruction mais de ce que l'on appelle, en langage bien barbare, un "pseudo-opcode" ou une "directive", c'est à dire une commande propre à l'assembleur utilisé (LISA 1.5 en l'occurrence), mais qui ne relève pas du langage machine et ne fera pas partie du programme objet résultant de l'assemblage. Les commandes de définition d'étiquettes que nous avons déjà évoquées (EQU, EPZ...) sont aussi des directives.
En l'espèce, ORG indique à l'assembleur à partir de quelle adresse il doit réaliser l'assemblage et calculer les sauts éventuels (un JSR à l'intérieur du programme par exemple, mais pas les branchements par BEQ ou BNE qui, rappelons-le, se font en valeur relative, en fonction d'un déplacement et non directement sur une adresse fixe).

Le programme fait peu de choses : il exécute un HOME, vous demande un caractère au clavier, trace un cadre autour de l'écran avec ce caractère, positionne ensuite le curseur au centre de l'écran pour vous demander un autre caractère avec lequel il trace un nouveau cadre sur le précédent et ainsi de suite (on ne peut en sortir que par RESET). Nous allons l'analyser en détail.

Ligne 2 : $FC58 est l'adresse où commence la routine du moniteur qui est exécutée lorsque vous faites un HOME en Applesoft. JSR $FC58 est donc équivalent à HOME (la routine se termine évidemment par RTS).
Ligne 3 : JSR $FD0C est équivalent à GET. La routine de saisie d'un caractère au clavier commence en effet à l'adresse $FD0C; elle stocke le caractère (le code-écran du caractère en fait) dans l'accumulateur.
Ligne 4 : on stocke le caractère à l'adresse $6, qui n'est pas utilisée par le système et reste à la disposition du programmeur. On donne à cette instruction l'étiquette B4 afin de pouvoir y effectuer un branchement ultérieurement.
Ligne 5 : on charge l'accumulateur avec le numéro de la première ligne de l'écran soit 0.
Ligne 6 : la routine du moniteur qui commence en $FBC1 calcule l'adresse de base d'une ligne de l'écran dont on donne le numéro par A. Sur une ligne, les caractères sont stockés de "adresse-base" à "adresse-base + $27" (de 0 à 39 en décimal, ce qui fait bien 40 caractères par ligne). La routine range cette adresse dans les mémoires $28 et $29.

Ligne 7 : Y=#$26 ou encore Y=38 en décimal, ce qui correspond à l'avant-dernier caractère de la ligne.
Ligne 8 : on remet le caractère entré au clavier dans A.
Ligne 9 : on le stocke à l'adresse "adresse-base de la ligne + 38"; il est ainsi affiché à l'écran (avant-dernier caractère de la première ligne).
Ligne 10 : Y=Y-1
Ligne 11 : si Y est différent de 0, on retourne à B0, afin d'afficher toujours le caractère sur la première ligne, de la droite vers la gauche. L'assembleur calculera lui-même le déplacement qui permettra de remonter à l'adresse où commencera l'instruction qui suit B0, ce dont on peut lui être reconnaissant.
Ligne 12 : Y=0 (le premier caractère de la ligne n'est pas encore affiché).
17 en hexadécimal vaut 23 en décimal : c'est le numéro de la dernière ligne de l'écran.
Ligne 13 : calcul de l'adresse-base de la dernière ligne.
Lignes 14 à 18 : on fait pour la dernière ligne comme pour la première.
Ligne 19 : Y=0 pour traiter la première colonne de l'écran.
Ligne 20 : on met $18 dans X au lieu de $17 afin de pouvoir placer le DEX avant l'opération d'affichage. Ainsi, le test sur la valeur de X (0 ou non) se fera après le traitement de la ligne 0 et non avant.
Ligne 22 : A = numéro de la ligne.
Lignes 23 à 25 : instructions réalisant l'affichage en première colonne (Y vaut toujours 0).
Ligne 26 : TXA a pour but de donner au "dernier résultat" testé par BNE la valeur de X, qui est bien celle qui nous intéresse ici.
Ligne 27 : on passe au caractère suivant (au-dessus) dans la même colonne, jusqu'à ce que la ligne 0 ait été traitée.
Ligne 28 : TYA pour donner au "dernier résultat" la valeur de Y. En fait, on veut savoir si la colonne que l'on vient de traiter est la première (0) ou la dernière ($27 = 39). Si c'est 0, on refait le travail avec $27 (lignes 30 et 31 qui renvoie à la ligne 20), sinon, le cadre est terminé et on se déplace en B3.
Lignes 32 à 34 : la mémoire $25 contient la position verticale du curseur; on la met à $C (12) et on fait un JSR à la routine qui exécute le VTAB ($FC22). Ceci est donc équivalent à VTAB 12.
Lignes 35 et 36 : $24 contient la position horizontale du curseur; on y met $14, ce qui équivaut à HTAB 20.
Ligne 37 : demande d'un caractère au clavier.

Lignes 38 à 39 : aucun caractère n'a 0 pour code-écran en mode normal et le test BNE réalisera donc systématiquement le branchement en B4 : le programme boucle !

Si vous voulez rentrer ce programme directement en hexadécimal, utilisez la récapitulation qui vous est donnée de la façon suivante :

Entrez en mode moniteur par CALL-151, puis tapez :
300: 20 58 ... 06 RETURN
308: A9 00 ... A5 RETURN
et ainsi de suite jusqu'au BA final. Retour au BASIC par 3D0G RETURN ou CTRL-C.

Vous pouvez, si vous le désirez, sauver la routine par BSAVE nom,A$300,L$4C (où $300 et $4C désignent respectivement l'adresse de début et la longueur du code à sauvegarder).
Pour l'exécuter, tapez CALL 768 RETURN.

Nous vous donnons également un petit programme en Applesoft qui fait le même travail que notre routine en assembleur. Essayez-le également et vous constaterez la chose suivante : on a le temps de voir le cadre se dessiner alors que, en langage machine, tout semble "instantané" (un cadre en remplace un autre sans que vous puissiez discerner une phase intermédiaire). Ceci confirme "de visu" que le langage machine permet de réaliser des traitements plus rapides que le BASIC et c'est bien la moindre des choses.

Mais, et la taille du programme, direz-vous ! Il ne faut pas se fier aux apparences : le programme source en assembleur peut vous paraître long (surtout si vous le tapez effectivement, sans être un spécialiste de la dactylographie), mais le code-objet n'occupe que 76 octets. Le "petit" programme Applesoft en prend pour sa part 112 ... Alors, rendez-vous au prochain article ! ■

LISTE D'ASSEMBLAGE LISA 1.5

```
0300            1          ORG $300
0300 2058FC     2          JSR $FC58
0303 200CFD     3          JSR $FD0C
0306 8506       4   B4     STA $6
0308 A900       5          LDA #0
030A 20C1FB     6          JSR $FBC1
030D A026       7          LDY #$26
030F A506       8          LDA $6
0311 9128       9   B0     STA ($28),Y
0313 88        10          DEY
0314 D0FB      11          BNE B0
0316 A917      12          LDA #$17
0318 20C1FB    13          JSR $FBC1
031B A026      14          LDY #$26
031D A506      15          LDA $6
031F 9128      16   B1     STA ($28),Y
0321 88        17          DEY
0322 D0F8      18          BNE B1
0324 A000      19          LDY #0
0326 A218      20   B20    LDX #$18
0328 CA        21   B2     DEX
0329 8A        22          TXA
032A 20C1FB    23          JSR $FBC1
032D A506      24          LDA $6
032F 9128      25          STA ($28),Y
0331 8A        26          TXA
0332 D0F4      27          BNE B2
0334 98        28          TYA
0335 D004      29          BNE B3
0337 A027      30          LDY #$27
0339 D0EB      31          BNE B20
0338 A90C      32   B3     LDA #$C
033D 8525      33          STA $25
033F 2022FC    34          JSR $FC22
0342 A914      35          LDA #$14
0344 8524      36          STA $24
0346 200CFD    37          JSR $FD0C
0349 AA        38          TAX
034A D0BA      39          BNE B4
```

RECAPITULATION

```
0300- 20 58 FC 20 0C FD 85 06
0308- A9 00 20 C1 FB A0 26 A5
0310- 06 91 28 88 D0 FB A9 17
0318- 20 C1 FB A0 26 A5 06 91
0320- 28 88 D0 FB A0 00 A2 18
0328- CA 8A 20 C1 FB A5 06 91
0330- 28 8A D0 F4 98 D0 04 A0
0338- 27 D0 EB A9 0C 85 25 20
0340- 22 FC A9 14 85 24 20 0C
0348- FD AA D0 BA
```

```
10  HOME : POKE 34,24
15  GET Z$: VTAB 1: HTAB 1: FOR I =
     1 TO 40: PRINT Z$;: NEXT : VTAB
     24: HTAB 1: FOR I = 1 TO 40: PRINT
     Z$;: NEXT
20  FOR I = 2 TO 23: VTAB I: HTAB
     1: PRINT Z$;: HTAB 40: PRINT
     Z$;: NEXT : VTAB 12: HTAB 20:
      GOTO 15
```

Un bug en virgule flottante

Christophe Chaumet nous a signalé ce bug, publié dans ACM SIGNUM Newsletter en avril 83. Les multiplications et divisions en virgule flottante de l'Apple sont sujettes à erreur dans certains cas. Ainsi, la mantisse contient 4 octets (32 bits); si, pour un nombre donné, les deux octets centraux de cette mantisse sont nuls et l'octet final non nul, il peut y avoir une erreur dans l'octet final pour les multiplications et divisions faites avec ce nombre.

Exemple :
Z = 0.5 +2^(-25)
W = 2*(Z/2)
On obtient en fait W = 0.5+2^(-26) comme résultat.

Par contre, si Z = 0.5 + 2^(-9) + 2^(-25), on trouve bien la même valeur pour Z et W, une fois que W a été défini par W = 2*(Z/2)

Apple a été informé de l'erreur il y a trois ans: hélas, ce problème est resté avec l'Apple //e.

Des POKEs à gogo

Roland Jost

L'interpréteur Applesoft est à l'étroit dans ses 10K de ROM et certaines de ses possibilités ne peuvent être exploitées que par l'emploi de POKEs à certaines adresses mémoire. D'autre part, il peut s'avérer nécessaire de tromper l'interpréteur qui est parfois un peu trop strict. Tout programmeur connaît les POKEs courants permettant de définir les limites de la fenêtre d'écran, de passer aux modes graphiques... D'autres sont moins connus. Cette liste rassemble tous ceux que j'ai pu trouver dans la littérature consacrée à l'Apple. Elle n'est certainement pas exhaustive et pourra être complétée par la suite.

Remarque : lorsqu'il s'agit de POKEr des valeurs sur 2 octets, X étant l'octet de poids faible, X+1 l'octet de poids fort, l'instruction suivante est utilisée :

POKE X+1,VAL/256 : POKE X,VAL −256*PEEK(X+1)

au lieu de :

V% = VAL/256 : POKE X,VAL−256* V%: POKE X+1,V%

Page zéro

- POKE 10,76: POKE 12,AD/256 : POKE 11,AD−PEEK(12)*256

Initialise la fonction USR. Un appel à USR provoquera un saut à un sous-programme machine commençant à l'adresse AD.

- POKE 32,X

Fixe la marge gauche de la fenêtre de texte (0<= X <= 39).

- POKE 33,X

Fixe la largeur de la fenêtre de texte (0<= X <= 39−PEEK(32))

- POKE 33,33

Affiche les instructions Applesoft sans mettre de marge. Nécessaire pour les corrections d'instructions.

- POKE 33,28

Permet un cadrage facile des REMarques.

- POKE 34,Y

Fixe le haut de la fenêtre de texte (0<= Y <= 23).

- POKE 35,Y

Fixe le bas de la fenêtre de texte (PEEK(34)<= Y <= 23).

- POKE 36,X

Fixe la position horizontale du curseur. A utiliser en remplacement de HTAB pour sortie sur imprimante si l'impression se fait sur plus de 40 colonnes.

- POKE 37,Y

Fixe la position verticale du curseur (0<= Y <=23).

- POKE 48,C

Sélectionne la couleur du tracé basse résolution (mode GR). 0=noir, 1=magenta, 2=bleu foncé, 3=pourpre, 4=vert foncé, 5=gris, 6=bleu, 7=bleu clair, 8=brun, 9=orange, 10=gris, 11=rose, 12=vert, 13=jaune, 14=incolore, 15=blanc.

- POKE 49,X

Sélectionne le MODE du moniteur.

- POKE 50,63

Sélectionne le mode d'affichage INVERSE.

- POKE 50,127

Sélectionne le mode d'affichage clignotant (mode FLASH).

- POKE 50,255

Retour au mode d'affichage normal.

- POKE 51,128

A placer en mode immédiat avant un ordre CONT, GOTO, GOSUB, sur la même ligne, lorsque la portion de programme à exécuter contient un ordre OPEN, READ, WRITE, APPEND ou POSITION. On évite ainsi le message NOT DIRECT COMMAND.

- POKE 55,AD/256 : POKE 54,AD −PEEK(55)*256

Changement de l'adresse du sous-programme de sortie de caractères. (Normalement $9EBD avec le DOS).

- POKE 57,AD/256 : POKE 56,AD −PEEK(57)*256

Changement de l'adresse du sous-programme d'entrée de caractères. (Normalement $9E81 avec le DOS).

- POKE 58,A1 : POKE 59,A2 : POKE 69,A3 : POKE 70,A4 : POKE 71,A5

Utilisation des adresses de sauvegarde de PC et des registres A,X,Y pour l'appel de sous-programmes du moniteur. En voici un exemple :

Conversion décimal en hexadécimal

10 INPUT "DEC=";D : POKE 71,D/256 : POKE 70,D − PEEK (71)*256 : POKE 59,249 : POKE 58,64 : POKE 49,0 : PRINT "HEX$=": : CALL−327

La valeur de D est POKEe dans les registres Y et X par l'intermédiaire des adresses de sauvegarde de ces registres ($47 ,$46). L'adresse du sous-programme d'affichage (PRNTYX - $F940) est POKEe dans les adresses de sauvegarde du PC ($3A, $3B). On met à zéro le MODE ($31). Le CALL−327 exécute un sous-programme restaurant les registres ($FEB9).

- POKE 60-67,X

Ces huit adresses permettent de transmettre des paramètres à certaines routines du moniteur : MOVE, VERIFY...

Exemple : déplacer de D1 à A1 une zone mémoire, commençant à l'adresse D1 et se terminant en D2 :

POKE 61,D1/256 : POKE 60,D1− PEEK (61)*256 : POKE 63,D2/256 : POKE 62,D2−PEEK (63)* 256 : POKE 67,A1/256 : POKE 66,A1 − PEEK (67)*256 : CALL −468

Le CALL−468 exécute la commande MOVE ($FE2C). Un CALL−548 (VERIFY $FE36) permettrait la comparaison de 2 zones mémoires.

- POKE 72,0

Met à zéro le registre d'état. Utilisé par exemple dans la routine de S.H.LAM (voir plus loin).

- POKE 82,213

A faire avant de sauver un programme sur cassette. Le programme s'exécutera automatiquement après un LOAD et le RUN n'est plus nécessaire.

- POKE 103,1 : POKE 104,AD/256 : POKE AD,0

Fixe à AD le début d'un programme Applesoft.

- POKE 106,X / 256 : POKE 105,X − PEEK (106) * 256

Fixe la valeur de LOMEM à la valeur décimale X. Identique à LOMEM:X.

- POKE 116,X / 256 : POKE 115,X − PEEK (116) * 256

Fixe la valeur de HIMEM à la valeur décimale X. Identique à HIMEM:X.

- POKE 118,0

Permet d'exécuter des instructions telles que INPUT, GET... en mode immédiat. L'adresse 118 ($76 - CURLIN) contient normalement $FF en mode direct et le numéro de ligne BASIC en mode programme. Exemple : POKE 118,0:INPUT " ";R$

- POKE 118,0 : POKE 43699,0

Permet un INPUT du clavier dans un fichier EXEC. Après l'INPUT, faire

un POKE 43699,1 (pour 48K). Toutes les instructions doivent être dans une même ligne.

- POKE 176,X / 256 : POKE 175,X – PEEK (176) * 256

Fixe l'adresse de fin d'un programme Applesoft. A utiliser, par exemple, lorsque l'on veut ajouter un programme binaire à la suite d'un programme Applesoft.

- POKE 214,255

Annule les commandes LIST, NEW, POKE, CALL et la commande SAVE du DOS. Empêche d'insérer ou d'effacer une ligne BASIC. Toute commande précitée fera exécuter le programme Applesoft en mémoire. A utiliser pour empêcher le listage et la modification de programmes. Pour revenir à la normale, passer en mode moniteur et faire "D6: 00".

- POKE 216,0

Annulation d'une instruction ONERR.

- POKE 228,X

Permet de changer le code de la couleur HGR. Noir=0, Vert=85, Violet=42, Blanc=127, Noir=128, Bleu=170, Orange=213, Blanc =255. Essayez toutes les autres possibilités pour X compris entre 0 et 255.

- POKE 230,32

Sélectionne la première page haute résolution ($2000-$3FF8) sans l'afficher. Toutes les instructions de tracé agiront sur cette page.

- POKE 230,64

Sélectionne la deuxième page HGR ($4000-$5FF8).

- POKE 230,96

Les instructions HGR agissent sur une 3ème page HGR "théorique" ($6000-$7FF8). Cette page ne peut être affiché par hardware, mais déplacée en bloc vers l'une des pages 1 ou 2.

- POKE 231,X

Permet de changer la valeur de SCALE (0-...).

- POKE 233,X / 256 : POKE 232,X – PEEK (233) * 256

Fixe l'adresse de début d'une table de formes.

- POKE 241,X

Fixe la vitesse d'affichage : SPEED = 256–X.

- POKE 243,0

Permet de sortir du mode FLASH après un <RESET> effectué en cours d'exécution de programme. Vous êtes dans cette situation lorsque, lors du balayage d'une instruction par la flèche à droite, le passage du curseur provoque l'affichage de caractères incohérents. Un POKE 243,0 rétablit une situation normale.

- POKE 243,32

Convertit les majuscules en minuscules lors d'une sortie sur imprimante.

- POKE 249,X

Fixe la valeur de ROT à X

Page 2

La page 2 ($200-2FF) sert de tampon (buffer) d'entrée des caractères frappés au clavier. La routine de S.H.LAM utilise des POKEs en page 2 pour installer en mémoire des sous-programmes assembleur. On peut aussi l'utiliser pour faire exécuter à partir du BASIC des instructions moniteur :

CO$=CO$ + " N D9C6G" : FOR Z=1 TO LEN(CO$) : POKE 511+Z, ASC(MID$(CO$,Z,1)) + 128 : NEXT Z : POKE 72,0 : CALL–144

Par exemple :

- CO$="2000<4000.5FF8M" copiera la page graphique 2 dans la page 1.
- CO$="300L" liste 20 instructions désassemblées à partir de l'adresse $300.
- CO$="C081 N C081 N D000<D000.FFFFM" copie la ROM dans la carte d'extension mémoire.

Page 3

- POKE 1009,X/256 : POKE 1008,X–PEEK(1009)*256

Change l'adresse de branchement en cas de BRK

- POKE 1011,X/256 : POKE 1010,X–PEEK(1011)*256

Modifie l'adresse de branchement en cas de RESET.

- POKE 1012,0

Provoque un démarrage à froid après un RESET

- POKE 1013,76 : POKE 1015,AD/256 : POKE 1014,AD–PEEK(1015)*256

Initialise & (AMPERSAND) pour un saut à un sous-programme dont le point d'entrée est à l'adresse décimale AD.

- POKE 1016,76 : POKE 1018,AD/256 : POKE 1017,AD–PEEK(1018)*256

Initialise CTRL-Y en mode moniteur pour un saut à un sous-programme situé à l'adresse décimale AD.

Pages 4-7

- POKE 1024 + X + 128*Y – 984*INT(Y/8),CA

Permet de POKEr directement en page 1 de TEXTE (écran) au point de coordonnées X et Y (0<= X <= 39 et 0<= Y <= 23). Le mode d'affichage dépendra de la valeur de CA :

0 <= CA <= 63 : MODE INVERSE
64 <= CA <=127 : MODE FLASH
128 <= CA <=191 : MODE NORMAL
192 <= CA <=255 : MODE NORMAL ou minuscules

Dans le cas d'une écriture en page 2 de TEXTE, remplacer 1024 par 2048. Attention, votre programme Applesoft ne doit pas débuter avant $C00.

Pages graphiques

Un POKE dans les adresses correspondant aux pages graphiques allumera un point sur l'écran. Vu l'organisation des pages HGR, il est peu pratique de POKEr directement des figures en page graphique.

DOS 48K

Toutes les adresses ci-dessous correspondent à un Apple 48K où le DOS est chargé entre $9600 et $BFFF. Pour des systèmes 32K ou 16K, il faut leur retrancher respectivement 16384 et 32768. Sauf indication contraire, ces adresses sont valables en DOS 3.2 et DOS 3.3.

- POKE 40407,234 : POKE 40408,234 : POKE 40409,234

La commande MON C,I,O n'est plus déconnectée lors d'un RESET.

- POKE 40514,52

Permet un démarrage du système (BOOT) sur un fichier binaire (valeur par défaut = 6 : fichier BASIC).

- POKE 40514,20

Permet un démarrage du système sur un fichier EXEC.

- POKE 40993,234 : POKE 40994,234 : POKE 40995,234

Permet de passer des commandes DOS (OPEN, READ...) en mode DIRECT, sans le message NOT DIRECT COMMAND.

- POKE 42424,192 : POKE 42432,193

Permet au DOS de reconnaître une carte ROM Applesoft ou Integer dans le port (slot) 4.

- POKE 43140,96 : POKE 43160,96 : POKE 43212,96

Ces trois POKEs annulent respectivement les commandes INIT, DELETE et RENAME du DOS en remplaçant la première lettre de chaque commande par un caractère non accessible au clavier de l'Apple II+.

Avec un POKE 214,255 en complément (voir plus haut), ceci constitue un bon moyen de protection de programmes contre les effacements ou interventions non autorisés.

Pour revenir à un DOS standard, faire "POKE 43140,73 : POKE 43160,68 : POKE 43212,82". L'effet du POKE 214 ne peut être annulé

qu'en mode moniteur (voir plus haut).

• POKE 43624,1
Sélectionne le lecteur de disquette 1.

• POKE 43624,2
Sélectionne le lecteur de disquette 2.

Le changement de lecteur peut se faire facilement par l'instruction "POKE 43624,3–PEEK(43624)"

• POKE 43626,X
Sélectionne le port (slot) du lecteur. X étant le numéro du port (6,5,4...).

• POKE 43699,0
Déconnecte un fichier EXEC. Le fichier reste ouvert jusqu'à la prochaine commande CLOSE, FP, INT ou MAXFILES.

• POKE 43699,1
Reconnecte le fichier EXEC.

• POKE 44452,0
Supprime la pause du CATALOG. Pour revenir à un affichage standard, taper "POKE 44452,22". D'autres valeurs donnent des résultats divers.

• POKE 44505,234 : POKE 44506,234
Les fichiers effacés (par DELETE) sont affichés lors d'un CATALOG.

• POKE 44596,255
Supprime la pause du CATALOG. Pour revenir à un affichage standard, taper "POKE 44596,206".
Un POKE 44596,0 entraîne un arrêt en mode moniteur et le CATALOG n'est pas affiché.

• POKE 44723,X*4 : POKE 44725,Y*4
A faire avant une initialisation de disquette. Les seules pistes accessibles pour la sauvegarde de fichiers seront celles comprises entre X et Y (inclus).

• POKE 44033,3 : POKE 44723,16 : POKE 44741,12 : POKE 44764,3 : POKE 46012,3
A faire avant une initialisation de disquette par INIT. Cette série de POKEs permet de placer le DIRECTORY et la VTOC sur la piste 3 au lieu de la piste 17. Cette disquette ne sera plus 'bootable' ni accessible par un DOS normal. Les programmes de copie (FID, COPYA) ne fonctionnent plus.
Dans un DOS standard, ces adresses contiennent respectivement les valeurs 17, 12, 68, 17, 17.

• POKE 49107,234 : POKE 49108,234 : POKE 49109,234
Evite, lors d'un BOOT à chaud, de recharger le langage complémentaire (Integer pour un Apple II+, ou Applesoft pour un Apple II) dans la carte langage ou la carte d'extension RAM).

BLOCK IOB

IOB (INPUT/OUPUT BLOCK) est utilisé par le sous-programme RWTS (Read Write Track Sector) pour l'accès à un secteur de la disquette. L'adresse de la table IOB utilisée par le DOS est donnée par :

IOB = 256*(PEEK(PEEK(996) + 256*PEEK(997)) + PEEK(PEEK(999) + 256*PEEK(1000))
normalement IOB = 47080 ($B7E8)

Pour modifier les paramètres de la table IOB, on pourra faire les POKES suivants :

• POKE IOB + 3,0
Met le numéro de volume à zéro. Donc accepte de lire la disquette quel que soit le numéro de volume. Sinon le numéro de volume doit correspondre au contenu de cette adresse.

• POKE IOB + 4,P
P = numéro de piste (0 - 34)

• POKE IOB + 5,S
S = numéro de secteur (0 - 15 en DOS 3.3 , 0 - 12 en DOS 3.2).

• POKE IOB + 9,A:256 : POKE IOB + 8,A–256*PEEK(IOB+8)
A = adresse du tampon en MEV.

• POKE IOB + 12,C
C = 0 : positionne la tête de lecture au début du secteur défini par P et S.
C = 1 : lecture du secteur P, S.
C = 2 : écriture du secteur P, S sur disquette.
C = 4 : formattage de la disquette. ATTENTION : détruit tout le contenu de la disquette.
Ne pas oublier de mettre la valeur de IOB dans les registres A et Y avant appel de RWTS (cf. manuel du DOS).

Entrées-sorties ($C000 $C0FF)

A ces adresses sont implantés des switchs (ou bascules) permettant de sélectionner une fonction d'entrée-sortie donnée.

• POKE 49168,0 (–16368)
Reprise de la lecture du clavier, le STROBE est remis à zéro.

• POKE 49184,0 (–16352)
Active la sortie magnétophone. Permet par exemple de sortir sur un amplificateur. Un deuxième POKE à la même adresse déconnecte la sortie.

• POKE 49200,0 : POKE 49200,0 (–16336)
Produit un 'CLIC' du haut-parleur.

• POKE 49232,0 (–16304)
Passage en mode graphique.

• POKE 49233,0 (–16303)
Passage en mode texte.

• POKE 49234,0 (–16302)
Choix du mode graphique non mixte.

• POKE 49235,0 (–16301)
Choix du mode mixte (4 lignes de texte en bas d'écran).

• POKE 49236,0 (–16300)
Choix de la page 1 (TEXTE OU GR OU HGR).

• POKE 49237,0 (–16299)
Choix de la page 2 (TEXTE OU GR OU HGR).

• POKE 49238,0 (–16298)
Basse résolution (mode GR).

• POKE 49239,0 (–16297)
Haute résolution (HGR).

Les commandes HGR et HGR2 effacent les pages graphiques. Pour passer à une page graphique sans effacement de son contenu, faire par exemple pour la page 2 :

POKE 49232,0 : POKE 49234,0 : POKE 49237,0 : POKE 49239,0

Pour basculer entre les pages 1 et 2 , exécuter l'instruction :

POKE 230, 96 – PEEK (230).

• POKE 49240,1 (–16296)
Met la sortie numérique 0 (patte 15 de la broche manette) à l'état "OFF" (5 volts).

• POKE 49241,0 (–16295)
Met la sortie numérique 0 en position "ON" (0,3 volts).

• POKE 49242,1 (–16294)
POKE 49243,0 (–16293)
Idem pour la sortie numérique 1 (patte 14).

• POKE 49244,1 (–16292)
POKE 49245,0 (–16291)
Idem pour la sortie numérique 2 (patte 13).

• POKE 49246,1 (–16290)
POKE 49247,0 (–16289)
Idem pour la sortie numérique 3 (patte 12).

• POKE 49248,0 (–16288)
Active l'entrée magnétophone. Un deuxième POKE à cette adresse désactive l'entrée.

• POKE 49280,0 (–16256)
Connecte la carte d'extension mémoire (BLOC 2) en mode lecture. Protection en écriture.

• POKE 49281,0 : POKE 49281,0 (–16255)
Déconnecte le BLOC 2 en mode lecture. Déprotège en écriture. La lecture de la ROM est possible.

• POKE 49282,0 (–16254)
Carte (BLOC 2) déconnectée en lec-

ture, protégée en écriture, c'est à dire retour au langage en ROM.

- POKE 49283,0 : POKE 49283,0 (−16253)

Le BLOC 2 de la carte d'extension est accesssible en lecture ET en écriture.

Les 12K du BLOC 1 sont accessibles de la même façon par les POKEs suivants:

POKE 49288,0 (−16252)
POKE 49289,0 (−16251)
POKE 49290,0 (−16250)
POKE 49291,0 (−16249)

Pour une carte d'extension 32K ou plus, la 2ème tranche de 16K est accessible par les POKES indiqués ci-dessous :

BLOC 2 : utiliser les POKES 49284 à 49287

BLOC 1 : utiliser les POKES 49292 à 49295.

Correction d'utilitaires

RENUMBER

L'utilitaire de renumérotation de lignes livré avec les disquettes Master DOS 3.2 et 3.3 comportait un "bug" qui peut être corrigé par les POKEs ci-dessous :

Après avoir chargé RENUMBER en mémoire par un LOAD, taper :

- POKE 4815,172 et POKE 4816,171 pour un DOS 3.2.
- POKE 4789,172 et POKE 4790,171 pour un DOS 3.3.

Sauver le programme modifié par SAVE RENUMBER.

Bibliographie

Tous ouvrages français consacrés à l'Apple II.
Revues : Pom's, L'Ordinateur Individuel, CALL A.P.P.L.E , NIBBLE , APPLE ORCHARD.

Conclusion

Tous ces POKES ont été testés avec DOS 3.3, 3.2, 3.2.1. A moins d'une erreur de transcription, ils devraient être corrects. Prière de signaler les erreurs éventuelles. Cette compilation ne tient certainement pas compte de nombreuses autres possibilités de modification du DOS (cf. les nombreux 'patches' possibles mais qui nécessitent plus que quelques POKES), ni des corrections éventuelles pour l'Applesoft chargé en carte langage. ■

Un programme de menu

Gilles Mauffrey

Ce programme en assembleur vous permet, en relation avec le programme de gestion de masques de G. Michel (POM'S 7), de réaliser très rapidement les menus de vos programmes BASIC.

Il permet de répondre à un menu soit par déplacement à l'aide des flèches -> ou <-, soit par réponse littérale (une lettre A,B,C...).
Ce programme est relogeable: vous pouvez donc l'intégrer très facilement dans tout programme BASIC. Avant de l'appeler, il est nécessaire de lui communiquer certaines informations par des POKE aux adresses correctes. Les informations concernées sont les suivantes :

- adresse 0: le numéro de la ligne de la première modalité de choix (entre 0 et 23).
- adresse 1: le numéro de la colonne à partir de laquelle se fera l'inversion des caractères (entre 0 et 39): cette colonne, pour chaque modalité de choix, doit contenir un caractère non blanc.
- adresse 2: le nombre de modalités de choix.
- adresses 36,37: positionnement du curseur en cas de réponse littérale.

Le petit programme de démonstration listé ci-après vous fournira les indications complémentaires pour la mise en oeuvre de la routine. ■

```
            ************
            *          *
            *DEMO  MENU*
            *          *
            ************

<A>  PREMIER CHOIX

<B>  SECOND CHOIX

<C>  TROISIEME CHOIX

<D>  QUATRIEME CHOIX

                    VOTRE CHOIX :

UTILISER LES -> , <- OU LA LETTRE VOULUE
```

Masque MASK DEMO

DEMOMENU

```
1   REM   PROGRAMME DE MENU
2   REM   (APPLESOFT)
3   REM   ==================
5 D$ =  CHR$ (4)
10  HOME : PRINT D$"BLOAD MASKDEMO
     ,A$8400,D1"
15  GOSUB 5000
20  POKE 0,7: POKE 1,6: POKE 2,4: POKE
     36,35: POKE 37,20
40  PRINT D$"BLOAD B:MENU,A$9500"
50  CALL 9 * 256 * 16 + 5 * 256
55  FOR J = 1 TO 600: NEXT
60 X =  PEEK (3): ON X GOSUB 1000,
     2000,3000
70  END
1000  HOME : PRINT "SP1"
1010  RETURN
2000  HOME : PRINT "SP2": RETURN
3000  HOME : PRINT "SP3": RETURN
5000  DATA  1024,1152,1280,1408,15
     36,1664,1792,1920,1064,1192,1
     320,1448,1576,1704,1832,1960,
     1104,1232,1360,1488,1616,1744
     ,1872,2000
5005  RESTORE
5010  FOR J = 1 TO 24: READ N: FOR
     K = 0 TO 39: POKE N + K, PEEK
     (8 * 256 * 16 + N + K): NEXT
     K: NEXT J
5020  RETURN
```

Programme MENU.TEXT DOS TOOL KIT

```
 1          ORG   $9000
 2 LIGNE    EQU   $00
 3 COLONNE  EQU   $01
 4 CHOIX    EQU   $02
 5 REP      EQU   $03
 6 LETMAX   EQU   $04
 7 TEMP     EQU   $05
 8 CH       EQU   $24
 9 CV       EQU   $25
10 BASE     EQU   $28
11 BASCALC  EQU   $FBC1
12 REPON1   LDA   CV
13          JSR   BASCALC
14          LDY   CH
15          LDA   #$5F
16          STA   (BASE),Y
17          LDA   #01
18          STA   REP
19          LDA   LIGNE
20          JSR   BASCALC
21          CLV
22          JSR   $FF58
23          BVC   INVERM
24          CLC
25          LDA   #'A
26          ADC   CHOIX
27          STA   LETMAX
28 PCL      LDA   $C000
29          BPL   PCL
30          BIT   $C010
31          CMP   #$C1
32          BCC   CTRL
33          CMP   LETMAX
34          BCS   PCL
35          BCC   LETTRE
36 CTRL     CMP   #$88
37          BNE   CTRT
38          BEQ   ARRIERE
39 CTRT     CMP   #$8D
40          BNE   CTRU
41 REVIEN   LDA   CV
42          JSR   BASCALC
43          LDY   CH
44          LDA   REP
45          ORA   #$C0
46          STA   (BASE),Y
47          RTS
48 CTRU     CMP   #$95
49          BNE   PCL
50          CLV
51          JSR   $FF58
52          BVC   EFFACE
53          INC   REP
54          LDA   REP
55          CMP   CHOIX
56          BCC   CAVA
57          BEQ   CAVA
58          LDA   #01
59 CAVA     STA   REP
60          CLV
61          JSR   $FF5B
62          BVC   PRECAL
63          BVS   PCL
64 INVERM   BVC   INVER
65 PRECAL   TSX
66          DEX
67          DEX
68          TXS
69          CLV
70          JSR   $FF58
71          BVC   CALINE
72          CLV
73          JSR   $FF58
74          BVC   INVER
75          BIT   $FF58
76          RTS
77 LETTRE   PHA
78          CLV
79          JSR   $FF58
80          BVC   EFFACE
81          PLA
82          SEC
83          SBC   #$C0
84          STA   REP
85          CLV
86          JSR   $FF58
87          BVC   PRECAL
88          BVS   REVIEN
89 ARRIERE  CLV
90          JSR   $FF58
91          BVC   EFFACE
92          DEC   REP
93          LDA   REP
94          BNE   CAVA
95          LDA   CHOIX
96          BNE   CAVA
97 EFFACE   TSX
98          DEX
99          DEX
100         TXS
101         LDY   #39
102 BINEF   LDA   (BASE),Y
103         ORA   #$C0
104         CMP   #$E0
105         BCC   STOB
106         SBC   #$40
107 STOB    STA   (BASE),Y
108         DEY
109         BPL   BINEF
110         BIT   $FF58
111         RTS
112 INVER   TSX
113         DEX
114         DEX
115         TXS
116         LDY   #40
117 INO     DEY
118         LDA   (BASE),Y
119         CMP   #'
120         BEQ   INO
121         EOR   #$E0
122         BEQ   INO
123 BINV    LDA   (BASE),Y
124         AND   #$3F
125         STA   (BASE),Y
126         DEY
127         CPY   COLONNE
128         BCS   BINV
129         BIT   $FF58
130         RTS
131 CALINE  TSX
132         DEX
133         DEX
134         TXS
135         LDX   REP
136         LDY   COLONNE
137         LDA   LIGNE
138         STA   TEMP
139         JSR   BASCALC
140 CALIN   DEX
141         BEQ   TROUVE
142 CALI    INC   TEMP
143         LDA   TEMP
144         JSR   BASCALC
145         LDA   (BASE),Y
146         EOR   #'
147         BEQ   CALI
148         BNE   CALIN
149 TROUVE  BIT   $FF58
150         RTS
```

RECAPITULATION

```
9000- A5 25 20 C1 FB A4 24 A9
9008- 5F 91 28 A9 01 85 03 A5
9010- 00 20 C1 FB B8 20 58 FF
9018- 50 51 18 A9 C1 65 02 85
9020- 04 AD 00 C0 10 FB 2C 10
9028- C0 C9 C1 90 06 C5 04 B0
9030- F0 90 4E C9 88 D0 02 F0
9038- 5D C9 8D D0 0E A5 25 20
9040- C1 FB A4 24 A5 03 09 C0
9048- 91 28 60 C9 95 D0 D2 B8
9050- 20 58 FF 50 51 E6 03 A5
9058- 03 C5 02 90 04 F0 02 A9
9060- 01 B5 03 B8 20 58 FF 50
9068- 04 70 B6 50 52 BA CA CA
9070- 9A B8 20 58 FF 50 68 B8
9078- 20 58 FF 50 42 2C 58 FF
9080- 60 48 B8 20 58 FF 50 1E
9088- 68 38 E9 C0 85 03 B8 20
9090- 58 FF 50 D9 70 A7 B8 20
9098- 58 FF 50 0A C6 03 A5 03
90A0- D0 BF A5 02 D0 BB BA CA
90A8- CA 9A A0 27 B1 28 09 C0
```

```
90B0- C9 E0 90 02 E9 40 91 28
90B8- 88 10 F1 2C 58 FF 60 BA
90C0- CA CA 9A A0 28 88 B1 28
90C8- C9 A0 F0 F9 49 E0 F0 F5
90D0- B1 28 29 3F 91 28 88 C4
90D8- 01 B0 F5 2C 58 FF 60 BA
90E0- CA CA 9A A6 03 A4 01 A5
90E8- 00 85 05 20 C1 FB CA F0
90F0- 0F E6 05 A5 05 20 C1 FB
90F8- B1 28 49 A0 F0 F3 D0 EE
9100- 2C 58 FF 60
```

Les logiciels de traitement de texte

Guy Lapautre

Avant d'aborder l'analyse de Gutenberg, il m'a semblé intéressant d'examiner la conception globale d'un système de traitement de texte sur micro-ordinateur. Pour cela, nous allons comparer à Gutenberg deux classiques : Magic Window et Apple Writer II.

Magic Window

Au lancement, Magic Window propose un menu permettant de s'orienter vers l'édition, l'impression, la gestion des disques, ...

Dans la plupart des cas, ce premier choix dirige vers un second menu. Par exemple, la fonction de gestion des disques proposera des options telles que sauvegarde ou relecture, annulation...

Quand on est à l'intérieur de l'option édition, le mode de fonctionnement normal est le remplacement : taper sur une touche, quel que soit l'endroit où l'on se trouve dans le texte, provoque le remplacement de l'ancienne frappe par la nouvelle et, si l'on est en fin de texte, l'ajout du nouveau caractère.

Si on désire insérer un ou des caractère(s) à l'intérieur d'un texte, il convient de se placer d'abord en mode insertion.

Enfin, les formats d'impression font partie intégrante du texte, si on choisit de les sauvegarder dans l'option "texte formaté".

Il s'agit là d'une des grandes originalités de Magic Window. Par ailleurs, les textes ainsi sauvegardés le sont en binaire. Ils tiennent moins de place sur disque que le même fichier en code ASCII, et les échanges disque sont plus rapides.

Apple Writer II

Au lancement, Apple Writer II se met en option édition, invitant à entrer directement un texte. Si ce n'est pas ce qu'on désire faire, il faut demander un menu (disques, impression, fonctions auxiliaires), ou effectuer directement une commande qui ne sort du mode édition que pendant son exécution (sauvegarde, lancement d'une impression).

Le mode de fonctionnement normal est l'insertion : taper sur une touche provoque l'insertion du caractère frappé avant celui sur lequel se trouve le curseur.

Si on désire procéder à un remplacement, il faut choisir ce mode particulier (qui ne permet d'ailleurs que l'utilisation des touches imprimables, à l'exclusion de toute clé de fonction ou de déplacement).

Les formats d'impression sont indépendants des textes : le même format peut être utilisé pour plusieurs textes, mais il faut demander chaque fois l'utilisation d'un format d'impression.

Par ailleurs, Apple Writer II possède deux fonctionnalités originales : une fonction Glossaire permettant l'usage très simple de mots ou petites phrases à caractère répétitif, et un langage associé permettant de programmer des commandes, le "WPL".

Gutenberg

Au lancement, Gutenberg propose une ligne de commande non commentée, qui permet de décider de l'utilisation d'un fichier (répertoire du disque, police de caractères, fichier texte,...).

La mise en position d'édition se fait par l'appel d'un fichier texte; il faut ensuite appeler la fonction choisie (par exemple : insertion). Pour changer de fonction, il convient de sortir du mode en cours (par ESC), puis de demander une nouvelle fonction.

Les formats d'impression sont totalement indépendants des fichiers texte. Le logiciel en propose une grande variété, et il est possible d'en créer d'autres. Un système de "macros" permet d'attacher un format standard à un texte. Un questionnement préalable à l'impression permet le choix d'un format non standard.

Une grande originalité de Gutenberg est la possibilité de dessiner ses propres polices de caractères : Gutenberg utilisant un écran graphique, toute fantaisie de représentation est possible : symboles mathématiques, sigles, logos,... En contrepartie, l'impression de ces caractères spéciaux est très lente.

Commentaires

Ces différences fondamentales de conception font qu'il est trompeur de donner une cote de commodité d'utilisation ou de facilité d'apprentissage à un logiciel de traitement de texte : certains préfèrent le menu à la ligne de commandes. D'autres préfèrent travailler normalement en mode remplacement (notamment ceux qui ont appris la dactylographie sur machine classique). D'autres enfin voudront avoir un texte prêt à l'emploi, y compris ses options d'impression.

On peut donc comparer les fonctionnalités. Les fonctionnalités de base sont à peu près les mêmes partout, seules variant les conditions d'utilisation. Par exemple, les déplacements dans le texte peuvent se faire uniquement grâce aux flèches de direction, ou utiliser des caractères de contrôle.

En revanche (et nous en avons déjà vu quelques exemples), il existe pour chacun des fonctions particulières qui peuvent sembler primordiales aux uns, compte tenu du type de travail qu'ils effectuent, inutiles aux autres qui les trouveront sans intérêt. ■

Gutenberg à l'essai

Guy Lapautre

Gutenberg est un logiciel de traitement de texte proposé par Gutenberg Software Limited, fonctionnant sur Apple II+ 64K ou //e.

Il existe en 2 versions : Gutenberg Junior (version testée ici) et Gutenberg Senior, présentant un certain nombre de fonctionnalités supplémentaires, dans des domaines qui seront indiqués au cours de l'analyse.

Gutenberg Jr est présenté sur une disquette double face contenant, outre le programme proprement dit, divers fichiers utiles et, sous forme de "démos", tout ce qui est nécessaire pour éditer le manuel sur imprimante.

Si certains usages (limités) s'accomodent d'imprimantes diverses, le logiciel est essentiellement écrit pour l'imprimante matricielle Apple. Un jeu important d'interfaces graphiques est proposé. L'interface PROM 1/2 pour Apple ne l'est pas, mais il suffit (nous l'avons découvert par tâtonnements...) de déclarer péremptoirement qu'on possède une interface Epson et tout marche bien (rien ne marche avec les autres interfaces proposées).

Plusieurs modes de fonctionnement sont possibles, en 40 ou 80 colonnes, en utilisant ou non l'écran graphique Haute Résolution pour les caractères spéciaux. Mais ATTENTION, l'option la plus avancée : 80 colonnes et écran graphique, nécessite une carte 80 colonnes ETENDUE (cette limitation fait l'objet de 2 lignes 1/2 "cachées" au fond d'un paragraphe).

Mode d'analyse

Nous évaluons les avantages et inconvénients de Gutenberg en fonction de ce qu'un utilisateur de traitement de texte est en droit d'attendre, c'est pourquoi nous allons d'abord tenter une définition de cet utilisateur : il ne peut pas se contenter des possibilités offertes par les traitements de texte classiques. Il est prêt à payer ces exigences par une complexité plus grande à l'usage. Par exemple :

- professeur de mathématiques voulant faire des polycopiés
- électronicien utilisant les symboles de sa profession
- styliste cherchant à associer des images, des logos...

Avantages et inconvénients sont présentés sous 7 têtes de chapitre :

1 - Approche du logiciel
2 - Les fonctions d'entrée de texte (édition)
3 - Le formatage des pages et paragraphes
4 - L'impression
5 - Les échanges disque
6 - La création de "polices de caractères"
7 - La documentation

1 - Approche du logiciel

Au lancement, le logiciel propose d'abord de choisir l'interface pour l'imprimante matricielle Apple que l'on va utiliser, puis d'indiquer le connecteur sur lequel cette carte est enfichée, enfin de choisir ou non le mode "démos".

En mode de fonctionnement normal, une ligne de commande apparaît en haut de l'écran, indiquant notamment le "mode écran" (haute résolution ou non) et le lecteur de disque en usage (évidemment le 1 lors du démarrage) et proposant de démarrer par "READY".

On peut alors décider de ce qu'on va faire, par exemple, changer de disque ou de mode écran, travailler en 40 ou 80 colonnes ou, après avoir pressé la touche "RETURN", éditer le catalogue du disque, lire un fichier, exécuter un programme utilitaire de gestion de disque...

Aucun mnémonique n'étant proposé, il y a donc au départ une dizaine de codes opératoires à connaître :

1 ou 2 - lecteurs de disques
4 ou 8 - 40 ou 80 colonnes
I ou O - travail en haute résolution ou non (hIgh; IOw)
6 - "re-boote" le système
Ctrl-D - mode "démos"
Ctrl-P - programme d'impression
RETURN - demande d'un nom de programme ou du catalogue

Nous retrouverons un peu tout au long des diverses fonctions un certain ésotérisme parfois gênant. C'est ainsi qu'en mode édition G est utilisé pour une recherche (comme GET et non pas F comme FIND) et X pour le remplacement d'un caratère par un autre (eXchange).

Avantages - L'écran n'est pas encombré par un menu. Une seule frappe suffit fréquemment pour réaliser une commande. L'exécution de cette commande, dans la mesure où il n'y a pas d'appel au disque, est très rapide.

Inconvénients - Il faut avoir une bonne mémoire pour emmagasiner tous les codes, ou un synoptique très bien fait (il manque cruellement dans la documentation).

Synthèse - Il SEMBLE que les auteurs de Gutenberg n'aient pas mis au premier rang de leurs préoccupations le souci de simplicité et de "convivialité", pour employer un terme à la mode. C'est clair et concis, c'est très abstrait. Il s'agit là d'une constatation et non d'un jugement.

2 - Les fonctions d'entrée de texte (édition)

Toutes les fonctions classiques sont là : déplacement du curseur dans le texte, insertion, effacement, remplacement, recherche par le contenu, déplacement de paragraphes, etc. Elles sont le plus souvent assorties d'un grand luxe de sous-fonctions.

A titre d'exemple, voici les sous-fonctions permettant de "détruire" tout ou partie d'un texte :

- K détruit le caractère sous le curseur
- DELETE détruit un mot
- DW aussi
- DL détruit une ligne

D suivi des flèches :

- gauche détruit le début d'une ligne
- droite détruit la fin d'une ligne
- haut détruit depuis le début de l'écran
- bas détruit jusqu'à la fin de l'écran

D suivi de :

- Ctrl-S détruit tout l'écran
- Ctrl-M marque la limite d'une destruction

Tout ce qui est ainsi détruit est irrécupérable.

Pour passer d'une fonction à une autre, il faut toujours "revenir à la case départ" par ESC. Soit par exemple la phrase "LOUI VAS A L"ECOLE" qu'on désire transformer en "LOUIS VA A L'ECOLE". Voici une séquence nonmale d'opérations :

Commande Y (insertion d'un caractère)
se placer derrière LOUI
taper S
ESC

Commande X (remplacement)
se placer sur "

taper
ESC

Se placer sur le S de VAS
K (destruction d'un caractère)

On peut imaginer plus simple !

ATTENTION, il y a un piège dans lequel l'auteur de ces lignes est souvent tombé : une séquence d'insertion doit se terminer par un retour aux caractères normaux, si on utilisait des caractères programmés. Si on a omis de le faire, aucune fonction ne reste disponible. Un "A" sur la ligne de commande prévient de cette situation. En sortir par "Pomme ouverte - RETURN".

Avantages - TRES larges possibilités dans de nombreux domaines (par exemple recherche dans le texte, recopies, remplacements). Ces possibilités ne seront probablement pas utiles à tous, mais chacun peut y puiser ce qui l'intéresse (c'est d'ailleurs écrit dans la notice).

Inconvénients - Mêmes remarques que pour le chapitre précédent, en notant qu'il ne s'agit plus d'une dizaine de codes, mais de plusieurs dizaines de sous-options (sans doute plus de 100 en comptant toutes les variantes).

Synthèse - Sans trop perdre des fonctionnalités du produit, il eût été possible de faire plus simple. La frappe de textes très longs sans "effets spéciaux" est lourdement pénalisée par le système d'édition adopté par Gutenberg.

3 - Le formatage des pages et des paragraphes

Là aussi, large panoplie de fonctionnalités, obligeant si on veut utiliser toutes les astuces, à une certaine jonglerie avec les "macros" : ce sont des commandes assez simples permettant la pagination, la présentation en 2 colonnes, l'édition de lettres, d'étiquettes, les différentes sortes de titrages hauts et bas, la numérotation en décimal ou en romain, l'indentation des paragraphes, leur numérotation, les sauts de lignes, ...

Des macros commandent aussi l'impression : types de caractères, gras ou maigres, espacement proportionnel ou non, avec des largeurs variées,..., ainsi que les débuts et fin de colonnes, de pages, enchaînement d'impressions,...

Il existe au moins une centaine de ces macros. Mais il existe aussi, pour traiter les cas courants, des formats standards sur la disquette. A noter que des formats plus nombreux sont proposés par Gutenberg Sr, ainsi que la possibilité de créer de nouvelles macros.

Les macros étant incluses dans le texte, le format d'impression, s'il est standard, fait donc partie de ce dernier et lui sera toujours attaché. Une précaution à observer : le format utilisé doit être disponible sur la disquette. Gutenberg le recherchera automatiquement.

Si en revanche vous désirez utiliser vos propres formats, l'indication devra en être donnée au moment de l'impression.

Avantages - Comme déjà vu, très larges possibilités, avec en plus le moyen de s'affranchir du maniement pas toujours simple des macros grâce aux formats préconstruits.

Inconvénients - Le nombre considérable de macros fait qu'il est totalement impossible de les connaître par coeur. L'absence d'un bon synoptique se fait sentir, à moins qu'on ne renonce aux possibilités offertes...

Synthèse - Parfois abondance nuit. Mais ne jetons pas la pierre à qui a voulu bien faire. Quel dommage qu'il n'y ait pas un résumé des commandes principales !

4 - L'impression

Gutenberg est écrit pour impression sur l'imprimante matricielle Apple (cf. début de l'évaluation). Il est possible, mais nullement certain, qu'il puisse fonctionner avec d'autres imprimantes. De même, une liste d'interfaces est proposée. D'autres sont possibles, la carte PROM 1/2 par exemple. Chaque utilisateur ne disposant pas d'une des configurations standard devra tâtonner, sans être sûr du succès.

L'impression d'un texte écrit en caractères "normaux" se fait sans problèmes particuliers. Veiller seulement à ce que le format d'impression soit présent sur la disquette.

En revanche, l'impression des caractères spéciaux (Alternate Character Set) se fait très lentement, car c'est une impression de GRAPHIQUE. Le passage, ligne par ligne, de l'impression texte à l'impression graphique, produit un effet curieux. Le papier monte et descend; il y a des retours arrière, mais strictement aucun problème n'a été rencontré. Le tout semble parfaitement fiable.

Attention : si on utilise dans un texte des caractères spéciaux, il faut veiller à ce que la police de caractères (FONT) utilisée pour l'édition se trouve bien sur le même disque que le texte au moment de l'impression.

On peut regretter qu'il ne soit pas possible d'imprimer un texte en mémoire sans passer par la séquence :

Sauvegarde sur disque (automatique)
Impression
Rappel (si on veut réutiliser le même texte)

sauf dans le cas particulier où on veut faire plusieurs impressions successives du même texte.

Il existe des macros permettant de "jouer" avec les caractères, y compris par exemple double soulignement ou italique, et d'autres pour programmer des impressions.

Signalons d'autre part une option "DUMP" qui permet l'impression totale d'un texte, Y COMPRIS les caractères de contrôle.

Avantages - Très grande souplesse et très grande fiabilité de l'impression, utilisant toutes les possibilités de l'imprimante matricielle Apple. On peut produire des textes de très bonne qualité (voir exemple en annexe).

Inconvénients - Tout juste peut-on citer l'impossibilité d'imprimer "en passant" un texte en mémoire.

Synthèse - L'impression est incontestablement un point fort de Gutenberg, et il faudrait être bien difficile pour ne pas apprécier les performances du logiciel dans ce domaine.

5 - Les échanges disque

Ils sont relativement nombreux mais généralement assez brefs. C'est sans doute la raison pour laquelle les auteurs proposent de faire une copie de la disquette Master, copie qui ne sera pas capable d'initialiser le système, mais qui pourra remplacer le Master après le "boot" initial.

La seule précaution notable à prendre consiste à s'assurer que l'ensemble de ce dont on a besoin pour éditer ou imprimer un texte est présent : texte lui-même, format et police de caractères.

Le passage d'un lecteur de disque à l'autre (2 lecteurs sont indispensables) se fait très simplement par la frappe du chiffre 1 ou 2 quand on est en position "READY".

Le système comporte un certain nombre de fonctionnalités intéressantes :

Tout d'abord, il est possible d'éditer ou d'imprimer le catalogue d'une disquette. Il est également possible de choisir un fichier (texte, format ou police de caractères) en pointant sur le catalogue.

Ensuite, un certain nombre de commande type DOS sont directement accessibles à partir de Gutenberg : initialisation, copie totale ou partielle, destruction, ...

Il n'y a pas lieu de faire de commentaires spéciaux à ce niveau.

6 - La création de caractères

C'est la principale originalité de Gutenberg par rapport aux traitements de texte classiques, celle qui fait que Gutenberg sera particulièrement apprécié par les utilisateurs dont les besoins sortent de l'ordinaire.

Il existe une option "SCREEN" à partir de laquelle on peut dessiner sur une matrice de 7*10 tout caractère, au sens le plus large du terme : alphabet étranger, symboles mathématiques, composants électroniques, structures chimiques, logos, ornementations diverses...

Si une matrice est trop petite, on peut en utiliser plusieurs. Chaque matrice sera désignée par une touche au clavier. La police de caractères spéciaux ainsi construite sera utilisée (et créée) en faisant précéder la touche frappée de "/" (pour 1 caractère), ou de "Pomme ouverte RETURN" (pour une suite de caractères). Dans ce dernier cas, on désamorcera l'utilisation des caractères spéciaux par un second "P.O. RETURN". Surtout, NE PAS L'OUBLIER (cf. remarque dans le chapitre édition).

Gutenberg Jr propose une de ces polices de caractères, comportant surtout des caractères accentués de format réduit... et d'esthétique discutable - voir l'exemple en annexe.

Gutenberg Sr est beaucoup plus riche dans ce domaine.

L'exemple annexé donne un aperçu du type de caractères et symboles qu'il est possible de créer et d'imprimer. A noter, pour les lecteurs réguliers de Pom's, que la méthode de création de caractères est très proche de celle décrite dans le Pom's 8, pages 21-23.

Avantages - Indiscutables, les avantages que comporte cette fonctionnalité font de Gutenberg un traitement de texte à part.

Inconvénients - Lenteur de l'impression dès qu'on est en présence de caractères spéciaux.

Synthèse - Le lecteur ne sera pas étonné de voir l'auteur particulièrement séduit. Mais un conseil s'impose : nous avons déjà vu à plusieurs reprises que les performances de Gutenberg se payaient au prix d'une certaine complexité, ce qui est bien normal. C'est vrai aussi ici. Gutenberg est sans doute le marteau-pilon pour écraser la mouche, s'il faut entrer uniquement du texte banal au kilomètre.

7 - La documentation

Gutenberg est vendu SANS DOCUMENTATION IMPRIMEE digne de ce nom : l'utilisateur imprime lui-même sa documentation en utilisant les démos proposées. On peut ou non apprécier une méthode qui certes réduit les coûts, mais qui complique la tâche de l'utilisateur (des heures de travail), et donne des résultats dont la qualité n'est pas comparable à celle d'un manuel bien fait.

Cette documention (en anglais) présente des lacunes, dont la principale est le manque total (déjà signalé) d'un synoptique résumant les commandes PRINCIPALES. Certes, de-ci de-là, l'utilisateur est avisé que telle option est indispensable, alors qu'on peut remettre à plus tard l'étude de telle autre. Mais comme c'est noyé dans l'ensemble du texte, la lecture n'en n'est pas facilitée.

On peut aussi lui reprocher de ne pas mettre en valeur la grande originalité que constitue la création de polices de caractères : très peu d'exemples sont présentés, alors que les formats et leurs macros tiennent de nombreuses pages.

Nous n'apprécions nullement cette documentation, s'agissant d'un produit qui n'est pas vendu à bas prix.

Conclusion

Gutenberg apparaît comme un traitement de texte présentant de nombreuses originalités et des fonctionnalités très variées.

Parmi ses faiblesses - Un maniement complexe, obligeant très souvent à "hacher" le travail pour passer d'une option à une autre - Le fait de n'accepter, en principe, que l'imprimante matricielle Apple - Une documentation totalement indigne d'un logiciel de cette classe.

Parmi ses forces - La possibilité pour l'utilisateur de créer ses propres "polices de caractères" - L'impression, surprenante au premier abord, mais très fiable, d'un mélange de texte et de graphique - La possibilité, malgré de nombreux tâtonnements au début, de trouver dans la panoplie des fonctionnalités offertes celles qui conviennent à un problème particulier.

En synthèse - Gutenberg est peut-être la ROLLS des traitements de texte sur Apple. Il fut un temps où la ROLLS était livrée chauffeur compris. Sans aller aussi loin, Gutenberg n'est sans doute pas un traitement de texte à mettre entre toutes les mains, mais en revanche il séduira tous ceux -nombreux-qui veulent sortir du standard et du banal.

Gutenberg, le second traitement de texte ?

■

EXEMPLE

Voici un exemple de frappe "au kilomètre", avec justification à gauche. On utilise pour ce faire la "MACRO" ■L.

Voici maintenant un exemple d'impression en italique et lignes justifiées à droite.

On peut aussi bien sur travailler en pleine justification: gauche et droite.

Quelques exemples de caracteres composés par l'utilisateur:

$$\int_{-\infty}^{\infty} (ax^3+bx^2+cx)\, dx$$

$$\sin^2(x_0+x_1+x_2)$$

☞

Ce texte est un exemple d'impression programmée sur 2 colonnes. De nombreux autres formats d'impression sont disponibles.

Pommesoft

Philippe Faure

Les programmes présentés ici permettent de charger l'Applesoft dans la carte langage, puis de le modifier. Les avantages que l'on peut retirer de l'implantation d'un BASIC modifié sont nombreux :

- possibilité de programmer en langue française (ou autre, bien sûr, selon les modifications apportées),
- possibilité de "traduire" en Pommesoft un programme déja écrit en Applesoft. Il suffit de sélectionner le Pommesoft et de charger le programme : la traduction est automatique,
- de même, la démarche inverse est possible, et ce de façon automatique grâce aux commandes &A et &P qui permettent respectivement de passer en Applesoft et en Pommesoft en cours de travail. Ce transfert s'effectue sans perte du programme (contrairement aux ordres FP et INT).
- modification de certaines routines de l'Applesoft (comme le "GARBAGE COLLECTION"...).

Le système impose un certain nombre de contraintes, et tout d'abord au niveau du matériel minimum requis :

- un lecteur de disquettes,
- sur un Apple II+, il faut avoir "une carte langage" d'au moins 16K,
- cette carte est incluse dans l'Apple //e, sur lequel a d'ailleurs été développé ce programme.

Par ailleurs, le programme EDITEURPS autorise la modification d'un ordre ou d'une commande et celle des messages d'erreurs. Cependant, on ne peut dépasser la longueur totale initiale des ordres, ni celle des messages. Si l'on veut remplacer INPUT par DEMANDE, il faut réduire un autre ordre (par exemple SHLOAD qui ne sert que sur cassette). En fait, il s'agit d'une restriction relativement peu contraignante car, les ordres devant toujours être tapés au clavier, la tendance serait plutôt à la contraction qu'au développement !

De plus, tous les ordres "cassette" peuvent être réduits à un caractère (si possible de contrôle du type DEL).

Les messages du SED ne sont pas modifiés mais il suffit d'utiliser un programme du style SPRECHEN SIE DOS paru dans Pom's (ou "DOS BOSS" par Beagle Brothers), ou de s'en inspirer, pour y parvenir.

Implantation et structure

Le système Pommesoft se compose de plusieurs éléments :

AMPERPOMME : programme qui utilise l'ampersand (&) et qui permet de changer de BASIC.
PSCREATEUR : utilise la routine de S.H.LAM pour mettre dans la carte RAM l'Applesoft trouvé en ROM.
EDITEURPS : édition, modification, écriture, sauvegarde du Pommesoft.
PS : "Pommesoft + Moniteur" chargé dans la carte langage, résultat des modifications.

AMPERPOMME

Il s'agit d'un programme qui a été publié dans NIBBLE EXPRESS No 2, page 143, et qui permet, grâce au vecteur &, de créer de nouvelles fonctions en assembleur pour le BASIC en affectant à chaque lettre de l'alphabet l'adresse d'un sous-programme en langage machine. Si la lettre choisie n'est pas assignée, il y a branchement sur SYNERR ($DEC9) pour affichage du message SYNTAX ERROR.
La version AMPERPOMME a été modifiée, car à l'origine, le BRUN impliquait une remise à zéro (vers SYNERR), ce qui obligeait à POKEr les adresse désirées. La version originale comportait en outre un test sur "(" pour le passage de paramètres, qui, lui aussi, a été supprimé (ce qui explique le RTS prématuré et les NOP). Sur la liste de désassemblage ci-après, on trouve à gauche l'AMPERPOMME et à droite la version originale, que l'on peut donc rétablir mais avec "&A(" en place de "&A" :

```
0300-A9 4C      LDA #$4C
0302-8D F5 03   STA $03F5
0305-A9 20      LDA #$20
0307-8D F6 03   STA $03F6
030A-A9 03      LDA #$03
030C-8D F7 03   STA $03F7
030F-60         RTS          A2=LDX
0310-32         ???
0311-A9 C9      LDA #$C9
0313-9D 4B 03   STA $034B,X
0316-A9 DE      LDA #$DE
0318-9D 4C 03   STA $034C,X
031B-CA         DEX
031C-CA         DEX
031D-10 F2      BPL $0311
031F-60         RTS
0320-8D 7F 03   STA $037F
0323-20 7D E0   JSR $E07D
0326-B0 03      BCS $032B
0328-20 C9 DE   JSR $DEC9
032B-20 B1 00   JSR $00B1
032E-EA         NOP          20=JSR
032F-EA         NOP          B8
0330-EA         NOP          DE
0331-AD 7F 03   LDA $037F
0334-38         SEC
0335-E9 41      SBC #$41
0337-0A         ASL
0338-AA         TAX
0339-BD 4B 03   LDA $034B,X
033C-B0 49 03   STA $0349
033F-BD 4C 03   LDA $034C,X
0342-8D 4A 03   STA $034A
0345-20 B7 00   JSR $00B7
0348-4C 80 03   JMP $0380
ADRESSES PAR LETTRE
034B-84 03      A=$0384      A
034D-C9 DE      B=SYNERR     B
034F-C9 DE                   C
0351-C9 DE                   D
0353-C9 DE                   E
0355-C9 DE                   F
0357-C9 DE                   G
0559-C9 DE                   H
035B-C9 DE                   I
035D-C9 DE                   J
035F-C9 DE                   K
0361-C9 DE                   L
0363-C9 DE                   M
0365-C9 DE                   N
0367-C9 DE                   O
0369-B0 03      P=$038D      P
036B-C9 DE                   Q
036D-C9 DE                   R
036F-C9 DE                   S
0371-C9 DE                   T
0373-C9 DE                   U
0375-C9 DE                   V
0377-C9 DE                   W
0379-C9 DE                   X
037B-C9 DE                   Y
037D-C9 DE                   Z
037F-D0
ROUTINES POUR A ET P
0380-8D 8D C0   STA $C080 (Carte)
0383-60         RTS
0384-8D 82 C0   STA $C082 (ROM)
0387-60         RTS
```

NDLR : cette routine ne constitue certainement pas le moyen le plus simple pour n'utiliser que les instructions &A et &P, mais nous vous la donnons néanmoins en l'état car elle pourra vous être utile pour d'autres applications.

PSCREATEUR

En utilisant la routine de S.H.LAM, ce programme fait une copie dans la RAM de l'Applesoft ROM et du moniteur. Ensuite, il charge "VOC" qui est une version légèrement modifiée des tokens Applesoft (tokens courants et non utilisés en graphique) et

des messages BREAK, REENTER... Vous trouverez en fin d'article la liste des tokens de VOC et leurs équivalents standards.
Les messages d'erreurs ne peuvent pas être chargés de cette manière car il faut modifier plusieurs adresses dans le code de l'Applesoft. Du fait de l'importance de l'Applesoft (42 secteurs), il est impossible d'en avoir une copie entière sur papier.

Il est bon de rappeler ici que les moniteurs de l'Apple II+ et de l'Apple //e ne sont pas identiques. Nous verrons plus loin comment créer concrètement un fichier PS, mais notez de suite qu'il est indispensable de faire "RUN PSCREATEUR" avant d'exécuter EDITEURPS pour une première manipulation sur votre propre matériel.

Structure de la table des tokens ($D●D●-$D25F)

Il est très facile de modifier cette table car tous les mots-clefs sont à la suite les uns des autres et l'interpréteur les "repère" par leur dernier caractère qui a le bit 8 "on".
Par exemple, END est codé "$45 $4E $C4" et $C4 est égal à $44 (D) + $80, $80 correspondant à 128 en décimal (cf EDITEURPS lignes 240 (lecture) et 3030 (écriture)).

Les messages d'erreurs sont d'un accès différent car l'Applesoft se sert du registre X pour pointer sur le début du message à écrire, en se décalant du début de la table des messages selon la valeur de X. Il faut donc rechercher dans l'Applesoft les références à "ERR●R" et donner les nouvelles valeurs de décalage (cf EDITEURPS lignes 3070 à 3240).

EDITEURPS

Ce programme permet de modifier les instructions et les messages de l'Applesoft. Nous en présenterons les fonctions dans l'ordre des menus qui vous sont proposés.

Premier menu

1 - CHARGEMENT DE PS
2 - SUITE DU PROGRAMME
3 - SORTIE -> BASIC

1 : chargement de PS et d'AMPERPOMME, puis passage au deuxième menu.
2 : passage au deuxième menu. C'est le choix que vous devez faire lors de la création de votre premier fichier PS.
3 : fin.

Deuxième menu

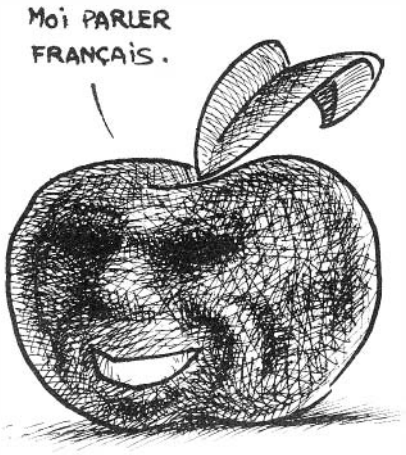
1 - COMMANDES APPLESOFT (modification des instructions)
2 - MESSAGES D'ERREURS (modification des messages)
3 - SORTIE -> BASIC

Après le choix des commandes ou messages, vous êtes amené au menu suivant.

Troisième menu

1 - LECTURE EN RAM
2 - LECTURE EN ROM
3 - SORTIE -> BASIC

1 : lecture des ordres ou messages stockés en RAM.
2 : lecture des tokens standards en DATA (lignes 50000 et suivantes). Pour une première édition, ce sont ceux que vous devez lire puisque, en principe, les modifications que vous apportez peuvent porter sur l'ensemble des instructions ou messages (les tokens "francisés" de VOC seront alors remplacés par vos propres adaptations). L'option 1 serait à utiliser ultérieurement pour une édition partielle sur un fichier PS déjà enregistré, ou encore pour ne remplacer que certains tokens du fichier VOC dès la première édition si la majorité d'entre eux vous convient.



Menu général

1 - VISUALISATION
2 - MODIFICATION
3 - ECRITURE
4 - SAUVEGARDE POMMESOFT
5 - SORTIE -> BASIC

1 : la visualisation permet de voir les tokens modifiés et leurs équivalents standards.
2 : les modifications se font dans un tableau de chaînes A$(150,1). Attention, aucune modification n'est effective tant qu'il n'y a pas eu écriture dans la RAM (option 3).
Vous pouvez traiter les ordres en séquence ou les appeler directement par leur numéro. Un RETURN à vide conserve l'ancienne définition et passe à l'item suivant. En cas de modification effective, la nouvelle définition est répétée; un RETURN la valide, mais toute autre touche annule la modification.
Le token numéro 40 est très important car il contient un complément pour respecter la longueur totale des tokens. Ce complément se fait automatiquement; en cours de modification on peut (il faut) le réduire au minimum.
Pour sortir de l'option "modification" avant la fin des tokens, il suffit de donner un numéro supérieur au nombre d'items.
3 : écriture dans la RAM du tableau A$ et modification des pointeurs s'il s'agit des messages d'erreurs.
4 : sauvegarde du contenu de la carte sous le nom de PS. ATTENTION : s'il existait une ancienne version de PS, celle-ci est écrasée. Pour éviter cela, ne pas oublier de la "renommer" après l'avoir chargée.

Remarque importante

Lors d'une modification des ordres, il ne faut pas oublier que l'interpréteur "regarde" du début de la table vers la fin. Aussi ne doit-on pas trouver dans un mot une suite de caractères équivalente à un autre mot qui lui succède. Je m'explique : j'ai eu des problèmes à cause de LISTE qui ne voulait pas fonctionner comme un simple LIST, mais j'avais par ailleurs remplacé READ par LIS. Comme READ se trouve avant LIST (et donc LIS avant LISTE), ma commande était interprétée "READ TE", ce qui me valait un beau message d'erreur !
Par contre, les ordres HGR et HGR2 fonctionnent parfaitement car HGR2 se trouve placé avant HGR dans la table et, de ce fait, est interprété avant lui.
Il faut aussi éviter de réduire certains tokens : si l'on remplace TO en A, par exemple, la variable A devient inutilisable en Pommesoft.

Revectoriser le FRE(0) : FRE(16)CONFIG

Ce petit programme montre comment, par l'intermédiaire de la RAM, on peut facilement modifier l'Applesoft.
En effet, lorsque la mémoire libre se réduit à moins d'un K octets, l'Applesoft procède automatiquement "au nettoyage", mais la routine employée (FRE(0)) est relativement lente.
Pom's a repris dans son numéro 2 un article de Randy Wissinton, paru dans Call A.P.P.L.E. de janvier 1981, qui proposait une routine de nettoyage-mémoire beaucoup plus rapide, que nous baptiserons FRE(16) pour la suite.
FRE(16)CONFIG charge cette routine et modifie l'adresse d'appel au FRE dans l'Applesoft en RAM. Il ne s'agit là que d'un exemple où la modification est assez "brutale" et loin d'être optimisée, mais les résultats sont éloquents : en utilisant le programme ci-dessous le chiffre 100 est inscrit au bout d'une minute environ

si l'on se trouve en Pommesoft modifié, alors qu'il faut plus de 2 minutes trente pour l'Applesoft standard !

```
10 DIM A$(10000)
20 A$(0) = "ABCDEFG"
30 FOR J = 1 TO 20: A$(J) = A$(0) + STR$ (I) + A$(0): NEXT : I = I + 1: PRINT I: GOTO 30
```

Un petit truc en passant...: pour redémarrer un programme BASIC par un GOTO (donc sans altérer les variables), tout en gardant le DOS connecté (et éviter l'ILLEGAL DIRECT ERROR), il faut taper "POKE 51,128: GOTO XXXXX" (XXXXX étant le numéro de ligne concerné). Cela permet, par exemple, de "renommer" le programme PS sur disquette et de redémarrer ensuite dans EDITEURPS sans rien effacer par un POKE 51,128 : GOTO 250

N.D.L.R. : si vous ne possédez pas la disquette d'accompagnement de ce numéro et si la frappe au clavier n'est pas votre loisir favori, notez bien que la saisie du code de VOC n'est pas indispensable au fonctionnement de Pommesoft (vous devrez simplement faire une première édition de PS sur l'ensemble des tokens standards, puisque qu'aucun d'entre eux n'aura été francisé dans la RAM).
Il faut dans ce cas enlever l'instruction "BLOAD VOC" dans le programme PSCREATEUR.

Si vous voulez, au contraire, saisir les codes hexadécimaux de VOC, vous pouvez utiliser la méthode suivante :

- Supprimer de PSCREATEUR les instructions "BLOAD VOC" et "BRUN AMPERPOMME" puis exécuter le programme ainsi réduit.
- Passer en moniteur par CALL −151 et saisir le code de VOC à partir de l'adresse $D000, exactement comme une récapitulation de programme-assembleur.
- Revenir au BASIC par 3D0G et taper "POKE 49280,0" pour sélectionner la RAM en lecture.
- Sauver VOC sur disquette par "BSAVE VOC, A$D000, L$260" et taper ensuite "POKE 49282,0" pour revenir au langage en ROM. ■

```
90  REM  PSCREATEUR
95  REM  ==========

100  POKE 49281, PEEK (49281):Y$ =
     "D000<D000.FFFFM N D9C6G": FOR
     I = 1 TO  LEN (Y$): POKE 511 +
     I, ASC ( MID$ (Y$,I,1)) + 128
     : NEXT : POKE 72,0: CALL  - 1
     44: PRINT : PRINT  CHR$ (4)"B
     LOAD VOC": PRINT  CHR$ (4)"BR
     UNAMPERPOMME"
```

```
90  REM  EDITEURPS
95  REM  ==========

100  POKE 49281, PEEK (49281)
110  TEXT : HOME : PRINT  TAB( 6)"
     CREATION DE BASIC FRANCAIS": POKE
     34,2
120 T = 10: PRINT : VTAB T: PRINT
     "1 - CHARGEMENT DE PS": PRINT
     : PRINT "2 - SUITE DU PROGRAM
     ME": PRINT : PRINT "3 - SORTI
     E -> BASIC"
130  PRINT : VTAB T + 7: PRINT "VO
```


MODEMS TELSAT

L'ACCES DE VOTRE MICRO ORDINATEUR AUX BASES DE DONNEES PAR LE RESEAU TELEPHONIQUE

TELSAT 440
Modem 300 bit/s full duplex

TELSAT 640
Modem 600/1200 bit/s full duplex

Ces produits font partie de la gamme complète de modems, multiplexeurs et équipements de réseaux de transmission de données TELSAT distribués par

SATELCOM international 69-71, rue du CHEVALERET 75013 PARIS
Tél. 584 14 75 Telex 204120F

```
     TRE CHOIX (2) ";: GET R$:R =
      VAL (R$):R = R * (R < 4): IF
     R = 0 THEN R = 2: IF R$ <  >
      CHR$ (13) THEN 130
132  PRINT R: IF R = 1 THEN  PRINT
     : PRINT  CHR$ (4)"BLOADPS": PRINT
      CHR$ (4)"BRUNAMPERPOMME"
135  IF R = 3 THEN 5000
136  POKE 49283, PEEK (49283)
150 X = DX: DIM A$(150,1),LL(30):Q
     $ =  CHR$ (34):AP = 0: HOME :
     T = 10: PRINT : VTAB T: PRINT
     "1 - COMMANDES APPLESOFT": PRINT
     : PRINT "2 - MESSAGES D'ERREU
     RS": PRINT : PRINT "3 - SORTI
     E -> BASIC"
160  PRINT : VTAB T + 7: PRINT "VO
     TRE CHOIX (1) ";: GET R$:R =
      VAL (R$):R = R * (R < 4): IF
     R = 0 THEN R = 1: IF R$ <  >
      CHR$ (13) THEN 160
165  PRINT R: IF R = 3 THEN 5000
170  PRINT : VTAB 2: IF R = 2 THEN
     DX = 0:FX = 16:DY = 53856:FY =
     53856 + 240: FOR I = 1 TO 107
     : READ A$: NEXT : PRINT  TAB(
     10)"MESSAGES D'ERREUR.": GOTO
     190
180 AP = 1:DX = 0:FX = 106:DY = 53
     456:FY = 53855: PRINT  TAB( 1
     1)"COMMANDES BASIC."
190 LO = FY - DY:L = 0:X = DX:Q$ =
      CHR$ (34): FOR I = DX TO FX:
      READ A$(I,0):L = L +  LEN (A
     $(I,0)): NEXT
200  HOME :T = 10: PRINT : VTAB T:
      PRINT "1 - LECTURE MESSAGES
     EN RAM": PRINT : PRINT "2 - L
     ECTURE MESSAGES STANDARDS EN
     ROM": PRINT : PRINT "3 - SORT
     IE -> BASIC"
210  PRINT : VTAB T + 7: PRINT "VO
     TRE CHOIX (1) ";: GET R$:R =
      VAL (R$):R = R * (R < 4): IF
     R = 0 THEN R = 1: IF R$ <  >
      CHR$ (13) THEN 210
220  PRINT R: IF R = 3 THEN 5000
230  IF R = 2 THEN  FOR I = DX TO
     FX:A$(I,1) = A$(I,0): NEXT : GOTO
     250
240 X = DX: PRINT : FOR 1 = DY TO
     FY:A =  PEEK (I): VTAB 20: PRINT
     X - 1"  "A$(X - 1 * (X > 0),1
     );: CALL  - 958: PRINT :A$(X,
     1) = A$(X,1) +  CHR$ (A - 128
      * (A > 127)):X = X + (A > 12
     8): NEXT
250  HOME :T = 10:H = 8: PRINT : VTAB
     T: PRINT  TAB( H)"1 - VISUALI
     SATION": PRINT : PRINT  TAB(
     H)"2 - MODIFICATION": PRINT :
      PRINT  TAB( H)"3 - ECRITURE"
     : PRINT : PRINT  TA8( H)"4 -
     SAUVEGARDE POMMESOFT": PRINT
     : PRINT  TAB( H)"5 - SORTIE -
     > BASIC"
260  PRINT : VTAB T + 11: PRINT  TAB(
     H)"VOTRE CHOIX ";: GET R$:R =
      VAL (R$): PRINT R: IF R < 1 OR
     R > 5 THEN 260
270 R =  - 1: ON  VAL (R$) GOTO 10
     00,2000,3000,4000,5000
1000  HOME :T = 10:H = 8: PRINT : VTAB
     T: PRINT  TAB( H)"1 - TOUT": PRINT
     : PRINT  TAB( H)"2 - BORNES":
      PRINT : PRINT  TAB( H)"3 - R
     ETOUR AU MENU GENERAL"
1010  VTAB T + 7: PRINT  TAB( H)"V
     OTRE CHOIX ";: GET R$: PRINT
     R$: IF R$ < "1" OR R$ > "3" THEN
     1010
1020 DE = DX:FI = FX: ON  VAL (R$)
      GOTO 1050,1030,250
1030  PRINT : VTAB T + 9: PRINT "V
     ALEUR DE DEBUT ("DX;: INPUT "
     ) : ";R$:DE =  VAL (R$) + DX *
     ( VAL (R$) = 0): PRINT : VTAB
     T + 9: PRINT "VALEUR DE DEBUT
      : "DE;"                    "
1040  PRINT : VTAB T + 11: PRINT "
     VALEUR DE FIN ("FX;: INPUT ")
      : ";R$:FI =  VAL (R$) + FX *
     ( VAL (R$) = 0): PRINT : VTAB
     T + 11: PRINT "VALEUR DE FIN
     : "FI;"                    "
1045  INPUT R$: IF R$ = "*" THEN  PRINT
     : PRINT "PR#1"
1050  INVERSE :C0 = 0:C1 = 0: FOR
     I = DE TO FI:C0 = C0 +  LEN (
     A$(I,0)):C1 = C1 +  LEN (A$(I
     ,1)): PRINT I: PRINT A$(I,0),
     C0: PRINT A$(I,1),C1: PRINT :
      IF I / 5 =  INT (I / 5) THEN
      GET R$
1060  NEXT : NORMAL : PRINT : PRINT
     "FIN DES ELEMENTS";: GET R$: PRINT
     : PRINT "PR#0": GOTO 1000
2000  HOME : PRINT : VTAB 5: PRINT
     "LONGUEUR "L"    MAXI"LO:I =
     R + 1
2010 T = 10: VTAB 10: PRINT "NUMER
     O (DE "DX" A "FX") DEFAUT = "
     I;: INPUT " : ";R$:R =  VAL (
     R$): IF R$ = "" THEN R = I
2015  IF R > FX OR R < 0 THEN 250
2020  PRINT : VTAB 5: PRINT "LONGU
     EUR ";L
2030  INVERSE : PRINT : VTAB 12: PRINT
     A$(R,0): PRINT A$(R,1): NORMAL

2040  INPUT R$: IF R$ = "" THEN 20
     00
2050  INVERSE : PRINT R$;: NORMAL
     : GET T$: IF T$ <  >  CHR$ (1
     3) THEN 2000
2060 L = L -  LEN (A$(R,1)) +  LEN
     (R$):A$(R,1) = R$
2070  GOTO 2000
3000  IF LO < L THEN  PRINT "ECRIT
     URE IMPOSSIBLE : MAXI < LONGU
     EUR";: POKE  - 16368,0: GET R
     $: GOTO 250
3005 A$(108,1) = "
                  "
```

```
3010  IF L < LO THEN A$(40,1) =  LEFT$
     ("@                          ,
                                   ",LO
      - L + 1)
3020 X = 0:I = DY
3030  FOR J = 1 TO  LEN (A$(X,1)):
     A = 128 * (J =  LEN (A$(X,1))
     ) +  ASC ( MID$ (A$(X,1),J,1)
     )
3040  POKE I,A:I = I + 1
3050  NEXT J: PRINT : VTAB 22: PRINT
     X" "A$(X,1);: CALL  - 958:X =
     X + 1: IF X =  < FX THEN 3030

3055  IF AP THEN 250
3060 C1 = 0: FOR X = DX TO FX:C1 =
     C1 +  LEN (A$(X,1)):LL(X + 1)
      = C1: NEXT
3070  POKE 57034,LL(1)
3080  POKE 55674,LL(2)
3090  POKE 57789,LL(3)
3100  POKE 56488,LL(3)
3110  POKE 57754,LL(4)
3120  POKE 59606,LL(5)
3130  POKE 54289,LL(6)
3140  POKE 58485,LL(6)
3150  POKE 55677,LL(7)
3160  POKE 57751,LL(8)
3170  POKE 57759,LL(9)
3180  POKE 60130,LL(10)
3190  POKE 58124,LL(11)
3200  POKE 56695,LL(12)
3210  POKE 58803,LL(13)
3220  POKE 58417,LL(14)
3230  POKE 55449,LL(15)
3240  POKE 58127,LL(16)
3250  GOTO 250
4000  PRINT : PRINT "UN MOMENT S.V
     .P...": PRINT : PRINT  CHR$ (
     4)"BSAVE PS,A$D000,L$3000": GOTO
     250
5000  POKE 49280,0
5010  TEXT : HOME : END
10020  FOR I = 56560 TO 56566
10025  PRINT  CHR$ ( PEEK (I))
10029  NEXT
10030  FOR I = 56544 TO 56556
10035  PRINT  CHR$ ( PEEK (I))
10039  NEXT
10040  FOR I = 54097 TO 54101
10045  PRINT  CHR$ ( PEEK (I))
10049  NEXT
10050  FOR I = 54105 TO 54106
10055  PRINT  CHR$ ( PEEK (I))
10059  NEXT
10060  FOR I = 54110 TO 54114
10065  PRINT  CHR$ ( PEEK (I))
10069  NEXT
50000  DATA  "END","FOR","NEXT","D
     ATA","INPUT","DEL","DIM","REA
     D","GR","TEXT","PR#","IN#","C
     ALL","PLOT","HLIN","VLIN","HG
     R2","HGR","HCOLOR=","HPLOT","
     DRAW","XDRAW","HTAB","HOME"
50010  DATA "ROT=","SCALE=","SHLOA
     D","TRACE","NOTRACE","NORMAL"
     ,"INVERSE","FLASH","COLOR=","
     POP","VTAB","HIMEM:","LOMEM:"
     ,"ONERR","RESUME","RECALL","S
     TORE","SPEED=","LET"
50020  DATA "GOTO","RUN","IF","RES
     TORE","&","GOSUB","RETURN","R
     EM","STOP","ON","WAIT","LOAD"
     ,"SAVE","DEF","POKE","PRINT",
     "CONT","LIST","CLEAR","GET","
     NEW"
50030  DATA "TAB(","TO","FN","SPC(
     ","THEN","AT","NOT","STEP","+
     ","-","*","/","^","AND","OR",
     ">","=","<","SGN","INT","ABS"
     ,"USR","FRE","SCRN(","PDL","P
     OS"
50035  DATA    "SQR","RND","LOG","
     EXP","COS","SIN","TAN","ATN",
     "PEEK","LEN","STR$","VAL","AS
     C","CHR$","LEFT$","RIGHT$","M
     ID$"
50040  DATA  "NEXT WITHOUT FOR","S
     YNTAX","RETURN WITHOUT GOSUB"
     ,"OUT OF DATA"
50050  DATA "ILLEGAL QUANTITY","OV
     ERFLOW","OUT OF MEMORY","UNDE
     F'D STATEMENT","BAD SUBSCRIPT
     ","REDIM'D ARRAY","DIVISION B
     Y ZERO","ILLEGAL DIRECT","TYP
     E MISMATCH","STRING TOO LONG"

50060  DATA  "FORMULA TOO COMPLEX"
     ,"CAN'T CONTINUE","UNDEF'D FU
     NCTION"
```

```
90  REM  FRE(16)CONFIG
95  REM  ==============

100  POKE 49281, PEEK (49281)
110  TEXT : HOME : PRINT  TAB( 9)"
     FRE(16) AUTOMATIQUE"
120  VTAB 7: PRINT "LE POMMESOFT D
     OIT ETRE CHARGE SOUS PEINEDE
     PLANTAGE !": PRINT : PRINT "C
     E PROGRAMME MODIFIE LA ROUTIN
     E          APPLESOFT 'GARBAG'
     EN Y METTANT L'ADRES-SE DE FR
     E(16)"
125 T = 16: PRINT : VTAB T: PRINT
     "1 - CHARGEMENT DE PS": PRINT
     : PRINT "2 - CONFIGURATION FR
     E(16)": PRINT : PRINT "3 - SO
     RTIE -> BASIC"
130  PRINT : VTAB T + 7: PRINT "VO
     TRE CHOIX (2) ";: GET R$:R =
      VAL (R$) * ( VAL (R$) < 4): IF
     R = 0 THEN R = 2: IF R$ <  >
      CHR$ (13) THEN 130
132  PRINT R: IF R = 1 THEN  PRINT
     : PRINT  CHR$ (4)"8LOADPS": PRINT
      CHR$ (4)"BRUNAMPERPOMME": GOTO
     100
135  IF R = 3 THEN  POKE 49280,0: TEXT
     : HOME : END
136  POKE 49283, PEEK (49283)
140  PRINT : PRINT  CHR$ (4)"BLOAD
     FRE(16),A$927C"
150  POKE 58500,76: POKE 58501,124
     : POKE 58502,146
```

RECAPITULATION AMPERPOMME

```
0300- A9 4C 8D F5 03 A9 20 8D
0308- F6 03 A9 03 8D F7 03 60
0310- 32 A9 C9 9D 48 D3 A9 DE
0318- 9D 4C 03 CA CA 10 F2 60
0320- 8D 7F 03 20 7D E0 80 03
0328- 20 C9 DE 20 81 00 EA EA
0330- EA AD 7F 03 38 E9 41 0A
0338- AA BD 48 03 8D 49 03 BD
0340- 4C D3 8D 4A 03 20 87 D0
0348- 4C 80 03 84 03 C9 DE C9
0350- DE C9 DE C9 DE C9 DE C9
0358- DE C9 DE C9 DE C9 DE C9
0360- DE C9 DE C9 DE C9 DE C9
0368- DE 80 03 C9 DE C9 DE C9
0370- DE C9 DE C9 DE C9 DE C9
0378- DE C9 DE C9 DE C9 DE 50
0380- 8D 80 C0 60 8D 82 C0 60
```

CODE DE VOC

```
D000- 6F D8 65 D7 F8 DC 94 D9
D008- 81 DB 30 F3 D8 DF E1 D8
D010- BF F3 98 F3 E4 F1 DD F1
D018- D4 F1 24 F2 31 F2 40 F2
D020- D7 F3 E1 F3 E8 F6 FD F6
D028- 68 F7 6E F7 E6 F7 57 FC
D030- 20 F7 26 F7 74 F7 6C F2
D038- 6E F2 72 F2 76 F2 7F F2
D040- 4E F2 6A D9 55 F2 85 F2
D048- A5 F2 CA F2 17 F3 88 F3
D050- 9E F3 61 F2 45 DA 3D D9
D058- 11 D9 C8 D9 48 D8 F4 03
D060- 20 D9 6A D9 D8 D9 6D D8
D068- EB D9 83 E7 C8 D8 AF D8
D070- 12 E3 7A E7 D4 DA 95 D8
D078- A4 D6 69 D6 9F D8 48 D6
D080- 90 E8 23 EC AF E8 0A 00
D088- DE E2 12 D4 CD DF FF E2
D090- BD EE AE EF 41 E9 09 EF
D098- EA EF F1 EF 3A F0 9E F0
D0A0- 64 E7 D6 E6 C5 E3 07 E7
D0A8- E5 E6 46 E6 5A E6 86 E6
D0B0- 91 E6 79 C0 E7 79 A9 E7
D0B8- 78 81 E9 7B 68 EA 7D 96
D0C0- EE 50 54 DF 46 4E DF 7F
D0C8- CF EE 7F 97 DE 64 64 DF
D0D0- 46 49 CE 50 4F 55 D2 53
D0D8- 55 49 56 41 4E D4 44 4F
D0E0- 4E 4E 45 45 D3 44 45 4D
D0E8- 41 4E 44 C5 53 55 D0 44
D0F0- 49 CD 4C 49 D4 47 D2 54
D0F8- 45 58 54 C5 50 52 A3 49
D100- 4E A3 43 41 4C CC 50 4C
D108- 4F D4 48 4C 49 CE 56 4C
D110- 49 CE 48 47 52 B2 48 47
D118- D2 48 43 4F 4C 4F 52 BD
D120- 48 50 4C 4F D4 44 52 41
D128- D7 58 44 52 41 D7 54 41
D130- 42 C8 45 43 52 41 CE 52
D138- 4F 54 B0 53 43 41 4C 45
D140- BD C0 54 52 41 43 C5 50
D148- 41 53 54 52 41 43 C5 4E
D150- 4F 52 4D 41 CC 49 4E 56
D158- 45 52 53 C5 46 4C 41 53
D160- C8 43 4F 4C 4F 52 BD 50
D168- 4F D0 54 41 42 D6 48 49
D170- 4D 45 4D BA 4C 4F 4D 45
D178- 4D BA 53 49 45 52 D2 52
D180- 45 50 41 52 D3 40 20 20
D188- 28 A0 C0 56 49 54 BD 53
D190- 4F 49 D4 56 41 53 2D 45
D198- CE 52 55 CE 49 C6 52 45
D1A0- 53 54 4F 52 C5 A6 47 4F
D1A8- 53 55 C2 52 45 54 55 52
D1B0- CE 52 45 CD 53 54 4F D0
D1B8- 53 45 4C 4F CE 57 41 49
D1C0- D4 C0 C0 44 45 C6 50 4F
D1C8- 48 C5 45 43 52 49 D3 43
D1D0- 4F 4E D4 4C 49 53 54 C5
D1D8- 45 46 46 41 43 C5 47 45
D1E0- D4 4E 45 D7 54 41 42 A8
D1E8- 56 45 52 D3 46 CE 45 53
D1F0- 50 A8 41 4C 4F 52 D3 41
D1F8- 04 4E 4F CE 50 41 D3 A8
D200- A0 AA AF DE 45 D4 4F D5
D208- 8E B0 8C 53 47 CE 45 4E
D210- D4 41 42 D3 55 53 D2 46
D218- 52 C5 45 43 52 A8 50 44
D220- CC 50 4F D3 52 41 C3 48
D228- 41 DA 4C 4F C7 45 58 D0
D230- 43 4F D3 53 49 CE 54 41
D238- CE 41 54 CE 58 45 45 C8
D240- 4C 4F CE 43 48 A4 56 41
D248- CC 41 53 C3 43 41 52 A4
D250- 47 41 55 A4 44 52 4F A4
D258- 4D 49 4C A4 49 44 A4 00
```

TOKENS Standards et Tokens de VOC

END	TEXT	HCOLOR=	TRACE	LOMEM:	IF	LOAD	NEW	+	(	SQR	LEN
FIN	TEXTE	HCOLOR=	TRACE	LOMEM:	IF	@	NEW	+	(	RAC	LON
1	10	19	28	37	46	55	64	73	82	91	100
FOR	PR#	HPLOT	NOTRACE	ONERR	RESTORE	SAVE	TAB(	-	SGN	RND	STR$
POUR	PR#	HPLOT	PASTRACE	SIERR	RESTORE	@	TAB(	-	SGN	HAZ	CH$
2	11	20	29	38	47	56	65	74	83	92	101
NEXT	IN#	DRAW	NORMAL	RESUME	&	DEF	TO	*	INT	LOG	VAL
SUIVANT	IN#	DRAW	NORMAL	REPARS	&	DEF	VERS	*	ENT	LOG	VAL
3	12	21	30	39	48	57	66	75	84	93	102
DATA	CALL	XDRAW	INVERSE	RECALL	GOSUB	POKE	FN	/	ABS	EXP	ASC
DONNEES	CALL	XDRAW	INVERSE	@	GOSUB	POKE	FN	/	ABS	EXP	ASC
4	13	22	31	40	49	58	67	76	85	94	103
INPUT	PLOT	HTAB	FLASH	STORE	RETURN	PRINT	SPC(	*	USR	COS	CHR$
DEMANDE	PLOT	TABH	FLASH	@	RETURN	ECRIS	ESP(	*	USR	COS	CAR$
5	14	23	32	41	50	59	68	77	86	95	104
DEL	HLIN	HOME	COLOR=	SPEED=	REM	CONT	THEN	AND	FRE	SIN	LEFT$
SUP	HLIN	ECRAN	COLOR=	VIT=	REM	CONT	ALORS	ET	FRE	SIN	GAU$
6	15	24	33	42	51	60	69	78	87	96	105
DIM	VLIN	ROT=	POP	LET	STOP	LIST	AT	OR	SCRN(	TAN	RIGHT$
DIM	VLIN	ROT=	POP	SOIT	STOP	LISTE	AT	OU	ECR(	TAN	DRO$
7	16	25	34	43	52	61	70	79	88	97	106
READ	HGR2	SCALE=	VTAB	GOTO	ON	CLEAR	NOT	)	PDL	ATN	MID$
LIT	HGR2	SCALE=	TABV	VAS-EN	SELON	EFFACE	NON	)	PDL	ATN	MIL$
8	17	26	35	44	53	62	71	80	89	98	
GR	HGR	SHLOAD	HIMEM:	RUN	WAIT	GET	STEP	=	POS	PEEK	
GR	HGR	@	HIMEM:	RUN	WAIT	GET	PAS	=	POS	PEEK	
9	18	27	36	45	54	63	72	81	90	99	

CALVADOS®

TOUTE LA BOURSE SUR L'ECRAN DE VOTRE ORDINATEUR PERSONNEL

- Cotations de 100 bourses dans 40 pays
- Gestionnaire de portefeuilles
- Lettres de Conseils ponctuels
- Documentation complète sur les sociétés
- Passage d'ordres par la messagerie
- et, bien entendu, tous les autres services Calvados

Documentation gratuite sur demande

Boîte Postale 21-07 75327 Paris Cédex 07
Tél. : (1) 705.09.04

02100 St.Quentin	COGNET	Mme. Metis	21, rue Victor Basch	(23)62.72.89
02200 Soissons	SOFIC	Mme Jullen	10.Place du Laon	(23)53.71.47
03200 Vichy	03 INFORMATIQUE	M. Rathueville	7. rue Voltaire	(70)31.74.00
06000 Nice	DSA INFORMATIQUE	M. Mouchon	5, Bd. Dubouchage	(93)85.15.96
06000 Nice	FNAC NICE	M.Couvreur	24, av. Jean Médecin	(93)92.09.09
06000 Nice	MCS	M. Abitbol	7, rue Dante	(93)96.50.55
06000 Nice	MICRO-MEDIA		27, rue Delille	(93)85.28.06
06000 Nice	SORBONNE INFORMATIQUE	M. Seyrat	40, rue Gioffredo	(93)5817.55
06150 Cannes la Bocca	ONDE MARITIME		28, Bd. du Midi	(93)47.4430
06400 Cannes	SIVEA CANNES	M. Corniflon	14, Bd. de la République	(93)39.29.09
06600 Antibes	ESPACE INFORMATIQUE	M. Irving	1, Chemin St.-Cloud	(93)7478.01
06700 St-Laurent-du-Var	COMPUTERLAND MOS	M. Ragazzi	Av. Léon Béranger	(93)07.61.12
06700 St-Laurent-du-Var	NOUVELLES GALERIES POINT MICRO	M. Guilbert	C.C. Cap 2000	(93)31.24.40
11000 Narbonne	R21 INFORMATIQUE	Mme. Perdriot	5, quai Vallière	(68)65.15.83
13005 Marseille	ELP INFORMATIQUE	M. Bocognano	20, rue Huguency	(91)94.91.13
13006 Marseille	COMPUTERLAND CPL	M. Giacobetti	1, av. de Corinthe	(91)78.02.02
13006 Marseille	ELECTRONIC SERVICE COMPUTER	M. Michelon	39, cours Lieutaud	(91)42.99.42
13006 Marseille	INTERNATIONAL COMPUTER 13		64, av. du Prado	(91)37.25.03
13007 Marseille	PROVENCE SYSTEM	M. Muniglia	74, rue Sainte/Le Saint-James	(91)33.23.33
13008 Marseille	MEDIATEC	M. Bouchoux	485, av. du Prado	(91)71.45.45
13008 Marseille	MICROMAG	M. Dewavrin	C.C. Mermoz/111, r. J. Mermoz	(91)71.79.13
13090 Aix-en-Provence	SOPROGA		Av. de l'Europe/Les Facultés B	(42)61.12.43
13100 Aix-en-Provence	ECO INFORMATIQUE AIX	M. Barental	Résid. Sextus/Bd. Victor Cousin	(42)27.11.48
13127 Vitrolles	FEEDER	M. Gauthron	Bastide Blanche RN 113	
13200 Arles	LUDO	M. Devred	27, rue de la République	(90)96.79.03
13231 Marseille	FNAC MARSEILLE		Centre Bourse	(91)91.30.62
14300 Caen	COMPUTERLAND CAEN	M. Desserteaux	12, rue St.-Pierre	(31)85.62.48
14490 Le Tronquay	NORMANDIE INFORMATIQUE	M. Cheval	Résidence La Tuilerie	(31)92.56.09
16000 Angoulême	L'HOMME	M. Lhomme	186, route de Bordeaux	(45)95.27.37
20121 Ferney-Voltaire	CIP	M. Sletzenhaar	43 bis, av. du Jura	(50)40.62.34
21000 Dijon	LASOBIKOR	M. Bikar	7, rue Monge	(80)30.09.70
21000 Dijon	OMG	M. Laurenti	20, rue Michelet	(80)30.12.70
21000 Dijon	SETTEM	M. Barbier	36, rue Jeannin	(80)66.16.43
24001 Périgueux Cédex	MPBI	M. Parinet	4, av. d'Aquitaine B.P. 92	(53)53.44.28
25000 Besançon	VAGNEUX S.A.	M. Joly	58 ter, Faubourg Rivotte	(81)81.12.56
25000 Montbéliard	MICRO ALPHA SOFT	M. Ravey	11, Impasse du Laquet	(81)97.16.46
26500 Bourg-les-Valence	ECA ELECTRONIQUE	M. Pons	22, Thannaron	(75)42.68.88
28000 Chartres	BIP INFO	M. Cabouro	58, rue Grand Faubourg	(37)21.74.51
29000 Quimper	LA BOUTIQUE INFORMATIQUE	M. Lemoelle	C.C. Rallye-Route de Benodet	(98)53.33.88
29000 Quimper	MATRICE DEVELOPPEMENT	M. Malté	2, Venelle de Kergos	(98)55.75.93
30000 Nîmes	ELECTRONIQUE TELE INFORMATIQUE	M. Aguilla	58, rue Pierre Semard	(66)36.02.52
30000 Nimes	MICRONIM	M. Belaïche	10, rue de la Trésorerie	(66)21.08.91
30120 Alès	ARCOMEL	M. Raffin	8 bis, rue Mistral	(66)30.05.62
31000 Toulouse	BUREAUMATIQUE	M. Tissot	4, Promenade des Capitouls	(61)2187.27
31000 Toulouse	FNACTOULOUSE	M. Claustre	1 bis, Promenade des Capitouls	(61)23.11.08
31000 Toulouse	SOUBIRON	M. Soubiron	9, rue Kennedy	(61)21.64.39
33081 Meriadeck Bordeaux Cédex	SIVEA BORDEAUX	M. Liegeois	Rue du Corps Franc Pommies	(56)96.28.11
34000 Montpellier	MICRO 34	M. David	7 Cours Gambetta	(67)92.91.23
34130 Valergues	CEBEA	M. Depeyroux	Route Nationale 113	(67)86.75.75
35000 Rennes	COMPUTERLAND BRETAGNE	M. Cador	13, av. du Mail	(99)54.47.12
35000 Rennes	X-MATIC	M. Hardy	161, av. du Gl. Patton	(99)38.31.80
35100 Rennes	DELTA MICRODIN	M. Preuveille	4, place Bretagne	(99)30.65.18
37000 Tours	AREI	M. Brossard	8, allée du Manoir	(47)64.69.93
37000 Tours	BOUTIN INFORMATIC	M. Boutin	36, rue Marceau	(47)20.51.83
37000 Tours	POLY SOFT	M. Caoudal	67, rue Michel Colombe	(47)66.66.49
37000 Tours	SELECTRON	M. Cazin	15, rue de Jérusalem	(47)20.80.70
38100 Grenoble	FNACGRENOBLE	M. Barbera	3 Grande Place	(76)09.46.63
38100 Grenoble	GAMMA INFORMATIQUE	M. Bragagnolo	9 Cours de la Libération	(76)96.30.05

Ces concessionnaires APPLE sont abonnés à CALVADOS. Demandez-leur une démonstration !

39100 Dole	GERMOND	M. Germond	12, av. Eisenhower	(84)72.49.11
40100 Dax	PLI		9, Cours Pasteur	
44013 Nantes	SIVEA-NANTES	M. Bacheller	21, Bd. Gabriel Guisth'au	(40)47.53.09
45000 Orléans	AGO	M. Autouin	48, Bd. Alexandre Martin	(38)54.45.13
45000 Orléans	AMC	M. Callier	13, rue des Minimes	(38)62.62.58
45000 Orléans	EUROPE ORDINATEUR	M. Poussegur	22, Bd. Alexandre Martin	(38)54.13.15
49000 Angers	INFORMATIQUE SERVICE	M. Fortier	42, rue Parcheminerie	(41)88.47.06
51100 Reims	ORGANIGRAMME	M. Coutant	16, rue Emile Zola	(26)88.51.13
53000 Laval	SLAD INFORMATIQUE	M. LeGrand	10, rue du Val de Mayenne	(43)49.25.45
54000 Nancy	PRECILAB	M. Bonnechere	96, rue Stanislas	(8)337.06.78
54000 Nancy	VLASTOSJEAN	M. Vlastos	143, rue Sgt. Blandon	(8)341.26.26
54404 Longwy Cédex	RANDOM	M. Lentz	18, rue Mercy/BP 410	(8)224.80.11
54520 Laxou	SEMITEC	M. Faber	69, rue de Mareville	(8)340.43.38
56000 Vannes	L'ORDINATEUR 56		38, Bd. de la Paix	(97)42.52.20
57000 Metz	FNACMETZ	M. Hochard	14, rue Tête d'Or	(8)736.16.22
57000 Metz	LA MICRO BOUTIQUE-ECONOMAISON	M. Le Besnerais	3, rue Paul Bezanson	(8)775.41.56
57110 Yutz	MICROSERVICE	M. Stutzmann	2 route de Thionville	
57800 Freyming Merlebach	CMI	M. Becker	3, place de la Gare	(8)781.14.89
59000 Lille	FNAC LILLE	M. Mortier	9, place du Général de Gaulle	(20)30.72.30
59000 Lille	SIVEA LILLE	M. Boisse	21 bis, rue de Valmy	(20)57.88.43
59150 Wattrelos	RYOCATTEAU		13, rue Edouard Vaillant	(20)75.92.23
59300 Valenciennes	MICROMEGA	M. Merchet	38, rue de Famars	(27)42.30.30
59700 Marcq en Barœul	MICRODATA INTERNATIONAL NORD	M. Williamson	920, av. de la République	(20)98.60.06
59800 Lille	MBDC	M. Bayart	172, rue Solferino	(20)57.91.87
59800 Lille	PAMP MICRO INFORMATIC	M. Paris	5, rue de Pas	(20)54.62.22
60100 Creil	HAPEL	M. Haentjens	2 bis, av. de l'Europe	(4)453.03.30
60108 Creil	QUENEUTTE	M. Queneutte	22 Bd. de la République	(4)425.04.26
61000 Alençon	TEMPS X	M. Courreau	42, rue de Lattre de Tassigny	(33)26.7998
62000 Arras	SICORFE	M. Delpierre/M. Recourt	18 bis, rue Lamartine	(21)23.08.56
62200 Boulogne-sur-Mer	DIF ELECTRONIQUE	M. Hagnere	71, rue du Camp de Droite	(21)30.75.68
63008 Clermont-Ferrand Cédex	NEYRIAL INFORMATIQUE	M. Neyrial	3 cours Sablon	(73)92.89.50
63014 Clermont-Ferrand Cédex	FLAGELECTRIC	M. Areste	21 du Brezet, 142, av. J. Mermoz	(73)92.13.46

suite page suivante ▶▶▶

SERVICE CALVADOS®

— LE RESEAU MICRO INFORMATIQUE —

64000 Pau	OBBO ADOUR	M. Zebrowski	14 Bd. Alsace Lorraine	(59)02.44.53
64100 Bayonne	LE CALCUL INTEGRAL	M. George	30, Bd.Alsace Lorraine-St.Esprit	(59)55.96.58
67000 Strasbourg	CILEC	M. Martin	18 Quai St.-Nicolas	(88)37.31.61
67000 Strasbourg	FNAC STRASBOURG	M. Rivet	22,place Kléber	(88)22.03.39
67000 Strasbourg	MICRO-MAT	M. Jung	30, rue Geiler	(88)60.68.68
67200 Strasbourg-Koenigshoffen	WALZ INFORMATIQUE		89 A Route des Romains	(88)29.54.55
68000 Colmar	EDSJL VOGEL	M. Vogel	8, rue Fischart	(89)80.10.86
68000 Colmar	FNAC COLMAR		1 Grande Rue	(89)23.32.12
68200 Mulhouse	FNAC MULHOUSE	M. Barremlin	1, place Franklin	(89)42.09.55
69002 Lyon	TERTIAL	M. Houot	19, rue Gentil	(7)839.16.66
69003 Lyon	ALPHA SYSTEMES	M. Barre	34, av. Maréchal de Saxe	(7)8608934
69003 Lyon	BIMP	M. Chastaing	20, rue Servient	(7)860.84.27
69004 Lyon	AB INFORMATIQUE	M. Asensio	18, rue Pailleron	(7)851.44.86
69006 Lyon	ALTI INFORMATIQUE	M. Combet	67, rue Vendôme	(7)894.60.56
69006 Lyon	ECO INFORMATIQUE		50 cours Vitton	(7)824.51.18
69007 Lyon	FNAC LYON	M. Thierry	62, av. de la République	(7)842.2649
71000 Macon	CLINIQUE ELECTRONIQUE	M. Bouquet/M. Damiani	369, rue de Paris	(85)38.20.84
72000 Le Mans	AESCULAPPLE	M. Besnard	4, rue de Richebourg	(43)24.97.80
72000 Le Mans	ASCI	M. Muzelle	115, rue Nationale	(43)84.2828
72000 Le Mans	OMB LE MANS	M. Coudreuse	42, rue Einstein	(43)24.99.40
74000 Cran Gevrier Annecy	SIGEA	M. Pollier	34, av. de la République	(50)57.02.80
74102 Annemasse	DSA MICRO	M. Domenjoz	15, rue Adrien Ligue	(50)38.31.40
75001 Paris	FNAC FORUM	M. Maisonnave	1, rue Pierre Lescot	(1)261.81.18
75001 Paris	NASA SEBASTOPOL		31, Bd. de Sébastopol	
75005 Paris	NASA MONGE	M. Roland	97, rue Monge	(1)656.52.82
75007 Paris	MVI	M. Vernet	27, rue Vanneau	(1)551.66.77
75007 Paris	NASA CEMP	M. Singer	28, av. de la Motte Piquet	(1)705.30.00
75008 Paris	AME	M. Bonnigal	172, Bd. Haussmann	(1)562.28.93
75008 Paris	EUROPE COMPUTER SYSTEMS	M. Lemoine/M. Guyonneau	16, rue Washington	(1)225.15.15
75008 Paris	FNAC ETOILE	M. Flandin	26, av. de Wagram	(1)766.52.50
75008 Paris	INTERSIS	M. Lallemand	2, rue Daru	(1)227.11.79
75008 Paris	ORDINATEUR INDIVIDUEL AFFAIRE	M. Hacegli	30, rue Cambaceres	(1)742.91.00
75008 Paris	POINT MICRO LA BOETIE	M. Anglade	16, rue de la Boétie	(1)265.89.35
75008 Paris	SIVEA	M. Cleenwerck/M. Bettiche	13, rue de Turin	(1)522.70.66
75009 Paris	COMPUTERLAND MOS 75	M. Giannetti	8, rue Bleue	
75009 Paris	JCR ELECTRONIQUE	M. Talar/M. Pattasso	58, rue Notre Dame de Lorette	(1)282.19.80
75009 Paris	LPS	M. Sokolski	46, rue Laffitte	(1)577.98.80
75009 Paris	NASA CEDL		45, rue Caumartin	
75009 Paris	SAVEMO	M. Vanfleteren	36, rue Laffitte	(1)770.46.44
75010 Paris	HELLO	M. Moreno	1, rue de Metz	(1)523.30.34
75010 Paris	SOFT MACHINE	M. Leconte	31, Bd. de Magenta	(1)240.85.00
75011 Paris	MID	M. Lamarre	51 bis, av. de la République	(1)357.83.20
75011 Paris	NASA SER		31, av. de la République	
75012 Paris	ELLIX	M. Bouvet	7, rue Michel Chasles	(1)307.65.58
75012 Paris	GMS	M. Laine	214, av. Daumesnil	(1)345.28.52
75013 Paris	NASA CEDM		C.C. Massena	
75014 Paris	NASA MONTELEC	M. Gaillard	88, av. du Maine	(1)656.52.82
75015 Paris	HI FI MADISON		127, rue St.-Charles	(1)578.81.16
75015 Paris	IEF	M. Deban	228, rue Lecourbe	(1)828.06.01
75015 Paris	ILLEL	M. Petit	143, av. Félix Faure	(1)554.97.48
75015 Paris	IMAGOL	M. Ratier	1-5, rue de Gutenberg	(1)5775.9.39
75015 Paris	KA	M. Rieul	212, rue Lecourbe	(1)533.13.50
75015 Paris	MICRO ASSISTANCE	M. Root	66, rue Castagnary	(1)530.05.28
75015 Paris	NASA DECOM		76, rue du Commerce	
75015 Paris	SIDEG	Mme. Lebail/M. Pewzer	170, rue St.-Charles	(1)557.79.12
75016 Paris	ANTIGONE	Mme. Muller	75 bis, rue Michel Ange	(1)743.13.41
75016 Paris	MICRODATA MDI	M. Perche	50, rue Raynouard	(1)525.81.64
75017 Paris	BSC INFORMATIC	M. Smadja	39, rue Cardinet	(1)763.95.33
75017 Paris	GAMIC	M. Galene	27, rue Guersant	(1)574.0292
75017 Paris	NASA GAD	M. Singer	46-48, av. de la Grande Armee	(1)574.59.74
75017 Paris	RANDOM PARIS	M. Lentz	75, Bd. Perreire	(1)227.59.20
76000 Rouen	ESPACE TEMPS REEL	M. Auriau	9, quai du Havre	(35)89.29.11
76000 Rouen	GUEZOULI INFORMATIQUE	M. Guezouli	39, quai du Havre	(35)8.9.55.66
76200 Dieppe	ELECTRODOM	M. Lequere	9, rue Lemoyne	(35)84.18.58
78000 Versailles	MICRORAMI	M. Ghysolin	17, av. de St.-Cloud	(1)021.30.15
78100 St-Germain-en-Laye	ORDI GESTION	M. Defie	13, rue des Louviers	(1)451.58.25
78117 Chateaufort	AVIRADIOAR INFORMATIQUE	M. Blassart	Aerodrome de Toussus le Noble	(1)956.22.20
78190 Trappes	PROBE INFORMATIQUE	Mme. Laporte	26, av. des Frères Lumières	(1)062.26.03
80000 Amiens	BUROTEC		38, rue de Beauvais	
80000 Amiens	LOGIC	M. Lecaille/M. Duport	18, rue des Augustins	(22)92.39.46
84000 Avignon	SYNERGIE INFO	M. Nasarre	71, Bd. Montclar	(90)86.52.32
85000 La Roche-sur-Yon	OMICRON INFORMATIQUE	M. Idler	2, place de la Résistance	(51)37.29.73
85105 Les Sables d'Olonne	IDEES INFORMATIQUES	M. Clouteau	Port Olonai B.P. 193	(51)95.19.47
86000 Poitiers	LISTE INFORMATIQUE	M. Bouat	8, rue Eperon	(49)41.43.86
87000 Limoges	MICROLIM	Mme. Almian	81, Bd. Gambetta	(55)34.10.12
90000 Belfort	FNAC BELFORT	M. Kachamiart	6, rue des Capucins	(84)21.00.15
91000 Evry	NASA SDESP		C.C. Evry II	
91700 Ste.-Geneviève	NASA GENELEC		96 route de Corbeil	
92086 Paris La Défense Cédex	POINT MICRO LA DEFENSE	M. Gotman/M. Vilotte	Tour Neptune	
92100 Boulogne Billancourt	MINIGRAPH MICRO INFORMATIQUE	M. Orsini	263, Bd. Jean Jaurès	(1)608.44.31
92120 Montrouge	SERAP MICRO INFORMATIQUE	M. Pacart/M. Benmaor	15, rue Louis Lejeune	
92130 Issy-les-Moulineaux	AGI	M. Nouailhac	4, rue Pierre Brossolette	(1)644.04.43
92200 Neuilly-sur-Seine	IMATIC		163, av. Charles de Gaulle	(1)747.11.26
92600 Asnières	NASA ADE		96, rue des Bourguignons	
92800 Puteaux	GEMINI INFORMATIQUE	M. Martin	52, rue Eichenberger	
94160 St-Mandé	MICRODATA ST-MANDE	M. Perche	5, rue Jeanne d'Arc	(1)808.64.35
95200 Sarcelles	MICROKEY95	M. Therry	Z.I., 34, av. de l'Escouvrier	(1)419.04.24
95526 Cergy Pontoise Cédex	ORGAMATIQUE	M. Poitier	9, chaussée J. César B.P. 304	(1)030.37.85
97200 Fort de France	POINT MICRO MARTINIQUE	M. Gallet de St.-Aurin	Centre Dillon	(596)71.49.6
97209 Fort de France	MICROTRAITEMENT	Mme de Reynal	C.C. Dillon	(596)73.91.4
97400 St-Denis de la Réunion	MICRO SYSTEMES SERVICES	M. Lebon	74, rue Labourdonnais	(262)21.62.5
98000 Monaco	MICROTEK	M. Hancette	2, Bd. Rainier III	(93)30.67.67
Maroc Casablanca	AMERICAN MICRO COMPUTERS	M. Housni	Av. des Far-Tours Habous	

Librairie dépositaire de la documentation technique Calvados :

LA NACELLE — 2, Rue Campagne-Première — 75014 PARIS - Tél. 322.56.46.

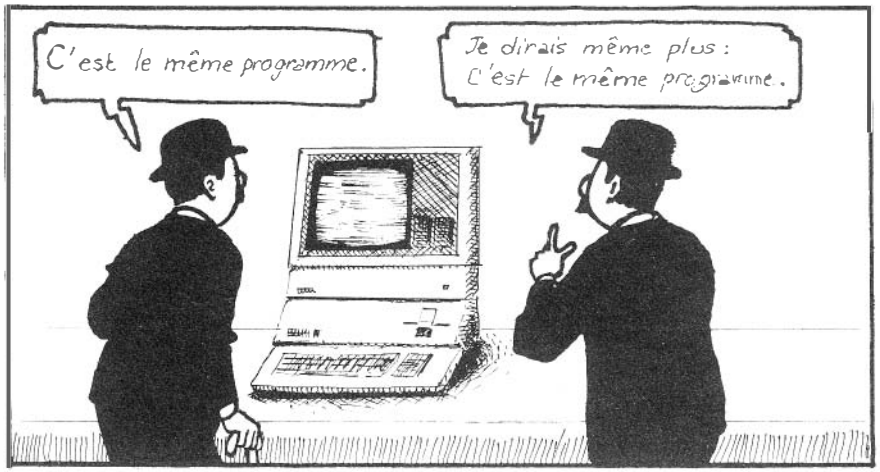
Comparaison de deux programmes en langage machine

Denis Sureau

Voici un utilitaire spécifique et rapide pour comparer deux versions d'un programme en langage machine (code hexadécimal), afin d'en identifier la plus récente ou de vérifier leurs différences.

Lorsque l'on a rentré le code de BCOMP, ou qu'on l'a chargé par un BLOAD à partir de la disquette d'accompagnement de ce numéro de Pom's, il faut l'exécuter une première fois par un "CALL 2061" avant sa sauvegarde (on appuie sur RETURN à la question "nom du programme" pour le stopper). On peut ensuite faire "SAVE BCOMP" et "RUN BCOMP" pour l'utiliser... J'ai bien dit "SAVE" et non "BSAVE" car, après ce premier CALL 2061, la routine se manipule comme un programme Applesoft.

L'explication de ce mystère tient à ce que les codes hexadécimaux qui figurent en tête de la routine sont la représentation-mémoire de la ligne Applesoft "10 CALL 2061". Lors de l'exécution, la position FPROG (fin de la routine) est stockée aux adresses 175-176 (pointeur de fin d'un programme Applesoft) et les trois derniers octets sont bien des 0, comme pour les programmes Applesoft.
Vous pouvez transposer directement cette petite astuce pour n'importe quel programme en langage machine dont l'origine est à l'adresse 2048 ($800): elle simplifie notamment le transfert des routines d'une disquette sur une autre.



Pour en revenir au vif du sujet, la routine charge les deux versions du programme à comparer à partir de l'adresse $1000 et compare chaque octet en se basant sur la longueur de la première version chargée. L'affichage des différences assure la conversion des adresses où elles se situent sur la base de l'adresse de départ normale du programme qui vous est demandée (en décimal positif) en début de traitement.
Le nombre de différences est indiqué en fin de comparaison et vous pouvez renouveler cette dernière sans charger les programmes en tapant "CALL 2064".

Pour imprimer les résultats, vous pouvez utiliser la procédure suivante :

- Exécuter une première fois BCOMP
- CALL −151
- PR#s (où s désigne le slot de l'imprimante)
- 8BDG

Les commentaires du programme source précisent le but de chaque procédure. Les adresses $6-$7 et $8-$9 pointent respectivement sur la position courante dans chaque version.
Une routine unique permet le chargement des deux versions en aiguillant sur chaque champ de stockage des noms réduits à 10 caractères en fonction du numéro de version donné par PRGNUM ($A1A).
La taille des programmes comparés ne doit pas dépasser 16K. ■

Programme BCOMP Assembleur BIG MAC

```
 1 *     B C O M P
 2 *
 3 *     (C) D. SUREAU
 4 *     (C) 1983 POM'S
 5
 6 *
 7 COUT      =     $FDED
 8 MAXI      =     $0008
 9 GETLN     =     $FD6F
10 LINPRT    =     $ED24
11 PRNTAX    =     $F941
12 PRBLNK    =     $F948
13 PRBYTE    =     $FDDA
14 STORADV   =     $FBF0
15 HOME      =     $FC58
16 TXTPTR    =     $00B8
17 INLIN2    =     $D52E
18 FRMEVL    =     $DD67
19 CRDO      =     $DAFB
20 GETADR    =     $E752
21 *
22 CTRL      =     $0018
23 CTRH      =     $0019
24 ADRL      =     $001A
25 ADRH      =     $001B
26
27           ORG   $800
28
29
30           HEX   000B080A008C32
31           HEX   303631000000
32           JMP   INTRO
33           JMP   START
34 INTRO     LDA   #FPROG
35           STA   175
```

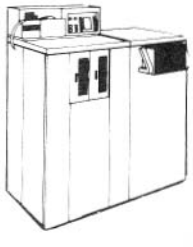
```
36          LDA  #>FPROG
37          STA  176
38          JSR  $3EA
39          JSR  HOME
40          JSR  ECRIT
41          HEX  8D8D8D00
42          LDA  #5
43          STA  $20
44          JSR  CRDO
45          JSR  TRAIT
46          JSR  ECRIT
47          INV  " B C O M P - D. SUREAU "
48          INV  "(C) 1983 "
49          HEX  8D00
50          JSR  TRAIT
51          LDA  #0
52          STA  $20
53          JSR  CRDO
54
55 *----------------
56 *CHARG PROGR 1  -
57 *----------------
58
59 CHARGE   LDA  #"1"
60          JSR  BLOADRTN
61
62 *-------------------
63 * CALCUL FIN PROGR -
64 *-------------------
65
66          CLC
67          LDA  43634
68          ADC  43616
69          STA  MAXI
70          LDA  43635
71          ADC  43617
72          STA  MAXI+1
73
74 *------------------
75 * CHARG PROGR 2   -
76 *------------------
77
78 CHARGE2  LDA  #"2"
79          JSR  BLOADRTN
80
81 *--------------------
82 * LECTURE ADRESSE  -
83 *--------------------
84
85          JSR  ECRIT
86          ASC  "ADRESSE NORMALE:"
87          BRK
88
89 * ENTREE D'UN NOMBRE PAR INPUT
90 * ET TRANSFERT EN $50-51
91
92          JSR  $FD6F
93          TXA
94          BEQ  INNUL
95          JSR  $D539
96          LDA  #0
97          STA  TXTPTR
98          LDA  #02
99          STA  TXTPTR+1
100         JSR  FRMEVL
101         JSR  GETADR
102         LDA  $50
103         STA  ADRL
104         LDA  $51
105         STA  ADRH
106         JSR  CRDO
107         JMP  START
```


LA PHOTOCOMPOSITION EN PROLONGEMENT DE LA MICRO-INFORMATIQUE

TRANSMETTEZ-NOUS VOS TEXTES
PAR TÉLÉPHONE
ou
DONNEZ-NOUS VOTRE DISQUETTE

Les textes de vos articles, catalogues, annuaires ou brochures saisis sur votre APPLE sont envoyés directement sur notre photocomposeuse.

Nous vous évitons ainsi, le coût et le temps de la saisie supplémentaire que nécessite le traitement traditionnel de la photocomposition avant l'impression des documents, si vous le désirez nous pouvons également nous charger de l'impression et du brochage.



NOTRE RÉFÉRENCE... LA REVUE POM'S

TELECOMPO 328.18.63

PHOTOCOMPOSITION BUREAUTIQUE TRANSMISSION DE DONNEES
GESTION DE FICHIERS MATERIEL DE TRAITEMENT DE TEXTES
13 et 15, avenue du Petit Parc 94300 VINCENNES

```
108
109 * PAR DEFAUT L'ADRESSE EST $1000
110
111 INNUL     LDA  #0
112           STA  ADRL
113           LDA  #$10
114           STA  ADRH
115
116 *--------------------
117 * INITIALISATION     -
118 *--------------------
119
120 START     LDA  #0
121           STA  CTRL
122           STA  CTRH
123           JSR  ENTETE
124           LDA  #0
125           STA  $6
126           STA  $8
127           LDA  #$10
128           STA  $7
129           LDA  #$50
130           STA  $9
131
132 *--------------------
133 * COMPARAISON        -
134 *--------------------
135
136           LDY  #0
137 COMPARE   LDA  $6
138           CMP  MAXI
139           BNE  COMPAR2
140           LDA  $7
141           CMP  MAXI+1
142           BCS  RECAP
143 COMPAR2   LDA  ($6),Y
144           CMP  ($8),Y
145           BEQ  INCCOMP
146
147 * AFFICHE L'ADRESSE (NORMALE)
148 * OU UNE DIFFERENCE EST VUE
149 * ET L'OCTET DANS LES DEUX
150 * VERSIONS DU PROGRAMME
151
152 DIFFRTN   LDA  $25
153           CMP  #21
154           BCC  AFFDIF
155           JSR  APPUYEZ
156           JSR  ENTETE
157
158 * AFFICHAGE DE L'ADRESSE
159
160 AFFDIF    LDA  #"$"
161           JSR  COUT
162           LDA  ADRL
163           CLC
164           ADC  $6
165           TAX
166           LDA  ADRH
167           ADC  $7
168           SEC
169           SBC  #$10
170           PHA
171           JSR  PRNTAX
172           INC  $24
173           PLA
174           JSR  LINPRT
175           JSR  PRBLNK
176
177 * AFFICHAGE DES OCTETS DIFFERENTS
178
179           LDY  #0
180           LDA  ($6),Y
181           JSR  PRBYTE
182           LDA  #24
183           STA  $24
184           LDA  ($8),Y
185           JSR  PRBYTE
186           JSR  CRDO
187
188 * DECOMPTE DES DIFFERENCES
189
190           INC  CTRL
191           LDA  CTRL
192           BNE  INCCOMP
193           INC  CTRH
194
195 * POSITION SUIVANTE
196
197 INCCOMP   INC  $6
198           INC  $8
199           LDA  $6
200           BNE  COMPARE
201           INC  $7
202           INC  $9
203           BNE  COMPARE
204
205 *-----------------
206 *     RECAP       -
207 *-----------------
208
209 * AFFICHE LE NOMBRE DE
210 * DIFFERENCES TROUVEES
211
212 RECAP     JSR  CRDO
213           LDA  $25
214           CMP  #21
215           BCC  RECAP2
216           JSR  APPUYEZ
217           JSR  HOME
218 RECAP2    LDA  CTRH
219           LDX  CTRL
220           JSR  LINPRT
221           JSR  ECRIT
222           ASC  " DIFFERENCE"
223           HEX  00
224
225           LDA  CTRH
226           BNE  PLURIEL
227           LDA  CTRL
228           CMP  #$02
229           BCC  EXIT
230 PLURIEL   LDA  #"S"
231           JSR  COUT
232 EXIT      JMP  $3D0
233
234 *------------------
235 * ROUTINES DIVERSES-
236 *------------------
237
238 ECRIT     PLA
239           STA  $0
240           PLA
241           STA  $1
242           LDY  #00
243 EC2       INC  $0
244           BNE  EC3
245           INC  $1
246 EC3       LDA  ($0),Y
247           BEQ  EC4
248           JSR  COUT
249           JMP  EC2
250 EC4       LDA  $1
251           PHA
252           LDA  $0
253           PHA
```

```
254            RTS
255
256 * EN-TETE
257
258 ENTETE     JSR  ECRIT
259            ASC  " ADRESSE     "
260 NOM1       HEX  A0A0A0A0A0
261            HEX  A0A0A0A0A0
262            HEX  A0
263 NOM2       HEX  A0A0A0A0A0
264            HEX  A0A0A0A0A0
265            HEX  8D00
266            LDA  #"-"
267            BNE  TRAIT1
268
269
270 TRAIT      LDA  #$2D
271 TRAIT1     LDX  #32
272 TRAITBCL   JSR  COUT
273            DEX
274            BNE  TRAITBCL
275            JMP  CRDO
276
277 * AFFICHE 'APPUYEZ SUR UNE
278 * TOUCHE' ET ATTEND QU'UNE TOUCHE
279 * SOIT PRESSEE
280
281 APPUYEZ    JSR  ECRIT
282            HEX  8D
283            ASC  "APPUYEZ SUR UNE TOUCHE"
284            ASC  " POUR POURSUIVRE"
285            HEX  8D00
286            BIT  $C010
287 CLVBCL     BIT  $C000
288            BPL  CLVBCL
289            BIT  $C010
290            JMP  HOME
291
292 *----------------------------
293 * CHARGEMENT D'UN PROGRAMME -
294 *----------------------------
295
296 BLOADRTN   STA  PRGNUM
297            JSR  ECRIT
298            ASC  "NOM DU PROGRAMME "
299 PRGNUM     HEX  00
300            ASC  " ;"
301            HEX  00
302            LDY  #21
303            LDA  #$A0
304 EFFACE     STA  NOM,Y
305            DEY
306            BPL  EFFACE
307
308 * ENTREE D'UN NOM DE PROGR.
309
310            JSR  GETLN
311            TXA
312            BNE  INPBCL
313            JMP  $3D0
314 INPBCL     LDA  $1FF,X
315            STA  NOM-1,X
316            DEX
317            BNE  INPBCL
318
319 * EFFACEMENT DES ZONES RESERVEES
320 * AUX NOMS DES 2 PROGRAMMES
321
322            LDX  PRGNUM
323            LDY  #10
324            LDA  #$A0
325 EFFBCL     CPX  #"2"
326            BEQ  EFFBCL2
327            STA  NOM1-1,Y
328            BNE  EFFBCL1
329 EFFBCL2    STA  NOM2-1,Y
330 EFFBCL1    DEY
331            BNE  EFFBCL
332
333 * TRANSFERT DU NOM DU PROGR.
334 * DANS UNE DES 2 ZONES
335
336            LDY  #0
337 TRANBCL    LDA  NOM,Y
338            CPX  #"2"
339            BEQ  TRANBCL2
340            STA  NOM1,Y
341            BNE  TRANS2
342 TRANBCL2   STA  NOM2,Y
343 TRANS2     INY
344            CPY  #10
345            BNE  TRANBCL
346
347 * AFFICHE 'BLOAD NOM-PROG,A$XXXX'
348 * PRECEDE DE CTRL-D ET OU XXXX
349 * EST $1000 OU $5000
350
351            JSR  ECRIT
352            HEX  8D84
353            ASC  "BLOAD"
354 NOM        HEX  A0A0A0A0A0
355            HEX  A0A0A0A0A0
356            HEX  A0A0A0A0A0
357            HEX  A0A0A0A0A0
358            HEX  A0A0
359            ASC  ",A$"
360            HEX  00
361            LDX  PRGNUM
362            CPX  #"2"
363            BEQ  BLOAD4
364            LDA  #$10
365            BNE  BLOAD5
366 BLOAD4     LDA  #$50
367 BLOAD5     LDX  #0
368            JSR  PRNTAX
369            JMP  CRDO
370
371            HEX  0000
372 FPROG      HEX  00
```

Récapitulation

```
0808- 00 0B 0B 0A 00 8C 32 38
0808- 36 31 00 00 80 4C 13 0B
0810- 4C B0 08 A9 A1 85 AF A9
0818- DA 85 B0 20 EA 03 20 58
0820- FC 20 73 09 8D 80 80 00
0828- A9 05 85 20 20 FB DA 20
0830- BC 09 20 73 09 20 02 20
0838- 03 20 0F 20 0D 20 10 20
0840- 2D 20 84 2E 20 13 15 12
0848- 05 01 15 20 28 03 29 20
0850- 31 39 38 33 20 80 80 20
0858- BC 09 A9 00 85 20 20 FB
0860- DA A9 B1 20 03 0A 1B A0
0868- 72 AA 6D 60 AA 85 0B A0
0870- 73 AA 6D 61 AA 85 0C A9
0878- B2 20 03 0A 20 73 09 C1
0880- C4 D2 C5 D3 D3 C5 A0 CE
0888- CF D2 CD C1 CC C5 BA 00
0890- 20 6F F0 BA F0 1F 20 39
0898- D5 A9 00 85 B8 A9 02 85
08A0- B9 20 67 D0 20 52 E7 A5
08A8- 50 85 1A A5 51 85 1B 20
08B0- FB DA 4C 8D 88 A9 00 85
08B8- 1A A9 10 85 18 A9 00 85
08C0- 1B 85 19 20 92 09 A9 00
```

```
08C8- 85 06 85 08 A9 10 85 07
08D0- A9 50 85 09 A0 00 A5 06
08D8- C5 08 D0 06 A5 07 C5 0C
08E0- 80 5A 81 06 01 08 F0 46
08E8- A5 25 C9 15 90 06 20 C9
08F0- 09 20 92 09 A9 A4 20 E0
08F8- F0 A5 1A 18 65 06 AA A5
0900- 18 65 07 38 E9 10 48 20
0908- 41 F9 E6 24 68 20 24 ED
0910- 20 48 F9 A0 00 81 06 20
0918- DA F0 A9 18 85 24 81 08
0920- 20 DA F0 20 F8 DA E6 18
0928- A5 18 00 02 E6 19 E6 06
0930- E6 08 A5 06 D0 A0 E6 07
0938- E6 09 D0 9A 20 F8 DA A5
0940- 25 C9 15 90 06 20 C9 09
0948- 20 58 FC A5 19 A6 18 20
0950- 24 ED 20 73 09 A0 C4 C9
0958- C6 C6 C5 02 C5 CE C3 C5
0960- 00 A5 19 D0 06 A5 18 C9
0968- 02 90 05 A9 D3 20 E0 FD
0970- 4C D0 03 68 85 00 68 85
0978- 01 A0 00 E6 00 00 02 E6
0980- 01 81 00 F0 06 20 ED FD
0988- 4C 7B 09 A5 01 48 A5 00
0990- 48 60 20 73 09 A0 C1 C4
0998- D2 C5 D3 D3 C5 A0 A0 A0
09A0- A0 A0 A0 A0 A0 A0 A0 A0
09A8- A0 A0 A0 A0 A0 A0 A0 A0
0980- A0 A0 A0 A0 A0 A0 8D 00
0988- A9 AD 00 02 A9 2D A2 20
09C0- 20 ED F0 CA 00 FA 4C F8
09C8- DA 20 73 09 8D C1 00 D0
0900- 05 D9 C5 DA A0 D3 05 D2
09D8- A0 D5 CE C5 A0 D4 CF D5
09E0- C3 C8 C5 A0 00 CF D5 D2
09E8- A0 D0 CF 05 D2 D3 D5 C9
09F0- D6 02 C5 8D 00 2C 10 C0
09F8- 2C 00 C0 10 F8 2C 10 C0
0A00- 4C 58 FC 8D 1A 0A 20 73
0A08- 09 CE CF CD A0 C4 D5 A0
0A10- D0 D2 CF C7 D2 C1 CD CD
0A18- C5 A0 00 A0 BA 00 A0 15
0A20- A9 A0 99 70 0A 88 10 FA
0A28- 20 6F F0 8A 00 03 4C D0
0A30- 03 8D FF 01 90 6F 0A CA
0A38- D0 F7 AE 1A 0A A0 0A A9
0A40- A0 E0 B2 F0 05 99 A0 09
0A48- D0 03 99 AB 09 88 00 F1
0A50- A0 00 89 70 0A E0 82 F0
0A58- 05 99 A1 09 D0 03 99 AC
0A60- 09 C8 C0 0A D0 EC 20 73
0A68- 09 8D 84 C2 CC CF C1 C4
0A70- A0 A0 A0 A0 A0 A0 A0 A0
0A78- A0 A0 A0 A0 A0 A0 A0 A0
0A80- A0 A0 A0 A0 A0 A0 AC C1
0A88- A4 00 AE 1A 0A E0 82 F0
0A90- 04 A9 10 D0 02 A9 50 A2
0A98- 00 20 41 F9 4C F8 DA 00
0AA0- 00 00
```

Deux produits nouveaux pour les lecteurs de Pom's

1. Le Disk Manager : le DOS en kit.

Présentation : une documentation d'une cinquantaine de pages et une disquette. Prix : 450 FF TTC port compris. Disponibilité : début avril.

Oeuvre de Dan Steevey, le Disk Manager est un programme utilitaire qui permet de recréer les commandes d'un DOS ou de redéfinir l'organisation d'une disquette, grâce à un jeu d'instructions qui en fait un langage simple d'accès au disque.

Il contient en seconde part un programme, particulièrement simple d'emploi, d'accès aux secteurs du disque, et qui offre pourtant des commandes évoluées.

Son but est de permettre la réalisation en Basic de programmes faisant directement appel au disque, en 10% du temps nécessaire pour écrire les mêmes programmes en assembleur.

Le Disk Manager comporte deux programmes en langage machine qui peuvent résider ensemble en mémoire, mais sont chargeables séparément :

- le Disk Organizer Manager (DOM) contient toutes les instructions de commande disque en 2Ko. Il peut être utilisé en mode direct ou à partir d'un programme Basic.
- le Disk Editor (DE) est une commande additionnelle, fonctionnant en mode direct ou en sous-programme, qui permet de charger, visualiser, éditer et réécrire des secteurs individuellement.

Utilitaires

La disquette contient également quatre programmes utilitaires élaborés avec ce langage d'accès au disque. L'analyse de ces utilitaires vous aidera à en maîtriser toutes les possibilités.

UTILI-DISQUE réunit des utilitaires classiques ainsi que des utilitaires semblables à ceux du CP/M ou du MS-DOS : reconstruction de disquette détruite, vérification, affichage de la carte d'occupation, ...

ULTRA-COPIE est la conversion en assembleur d'un algorithme de copie de disquettes mis au point en Basic (et présenté dans la documentation) : c'est un programme de "backup" de fichier très simple d'emploi et le plus rapide actuellement sur Apple II (à notre connaissance).

EDICAT offre le moyen d'éditer le catalogue d'une disquette afin d'en améliorer la présentation, de classer les programmes, d'ajouter des titres, un en-tête, ...

MULTI-DISQUE construit une base de données de tous vos programmes : il suffit de placer une disquette dans le lecteur pour incorporer son catalogue au fichier. Une routine de tri inédite (à deux tableaux simultanés) réalise un classement instantané.

NDLR - Le Disk Manager est à la fois un excellent outil et un moyen très efficace de comprendre le DOS et d'en utiliser toutes les ficelles à partir du Basic. Chacun peut alors se construire simplement des utilitaires personnalisés.

2. DBSTAG - Utilitaires pour DBASE II.

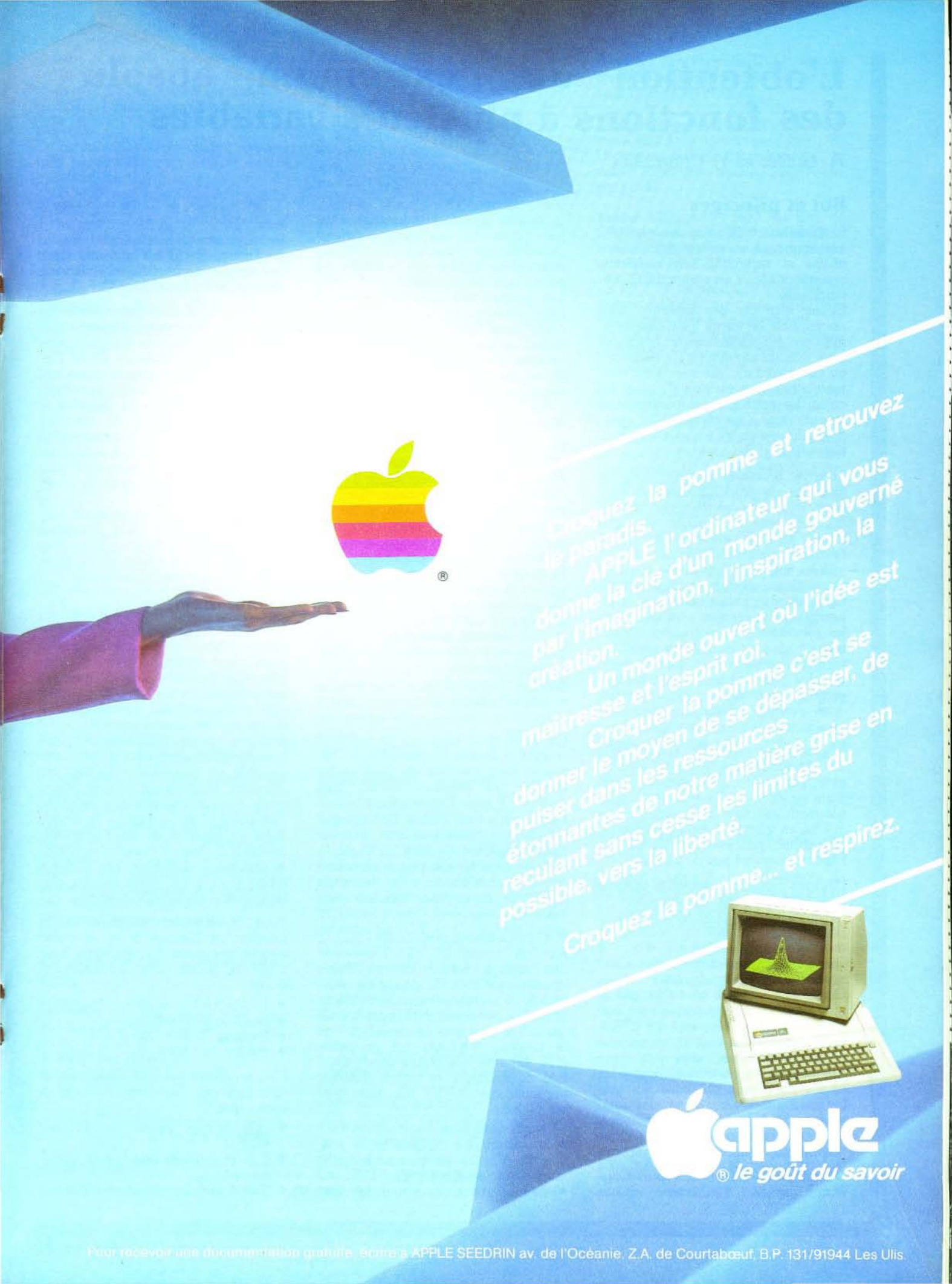
Présentation : une disquette CP/M comportant les programmes et une documentation détaillée imprimable. Prix : 450 FF TTC port compris.

DBSTAG est un logiciel associé à DBASE II et écrit pour Apple II avec carte Z80, pour fonctionnement sous CP/M et carte 80 colonnes et minuscules à l'écran. Ce logiciel permet d'effectuer des calculs statistiques souvent utiles en gestion : moyenne et écart-type d'une série de valeurs, construction d'histogrammes, calculs de corrélation (covariance, coefficient de corrélation, le cas échéant ordonnée à l'origine et pente de la droite de régression), analyse en déciles et analyse de Pareto. Ce logiciel n'est pas destiné à des statisticiens, mais à des gestionnaires.

Les résultats, fournis sous forme chiffrée, sont assortis de représentations semi-graphiques, en caractères imprimables par toute imprimante 80 colonnes ou plus.

DBSTAG peut exploiter, sans aucune transformation ou limitation, toute base de données construite sous DBASE II. Il peut être utilisé seul, ou comme sous-programme d'un programme écrit par l'utilisateur. Simple à comprendre et à apprendre, il est livré avec un exemple complet qui illustre parfaitement ses possibilités. ■

L'inspiration
ROUX, SEGUELA, CAYZAC & GOUDARD


Croquez la pomme et retrouvez le paradis.
APPLE l'ordinateur qui vous donne la clé d'un monde gouverné par l'imagination, l'inspiration, la création.
Un monde ouvert où l'idée est maîtresse et l'esprit roi.
Croquer la pomme c'est se donner le moyen de se dépasser, de puiser dans les ressources étonnantes de notre matière grise en reculant sans cesse les limites du possible, vers la liberté.
Croquez la pomme... et respirez.
®
apple
® le goût du savoir
Pour recevoir une documentation gratuite, écrire à APPLE SEEDRIN av. de l'Océanie, Z.A. de Courtabœuf, B.P. 131/91944 Les Ulis.

L'obtention de l'extremum absolu des fonctions à plusieurs variables

A. Guillez et Y. Cherruault

But et principes

Le problème traité dans cet article est très important en informatique scientifique et technique, en recherche opérationnelle et en comptabilité prévisionnelle.
Devoir identifier des fonctions dont on connaît la forme mathématique, mais pas les coefficients, et employer pour cela un algorithme de minimum ou de maximum, est une tâche courante dans tout bureau d'étude. En outre, les méthodes de minimisation sont d'usage pour résoudre les systèmes différentiels et aux dérivées partielles (Ritz et Rayleigh).

Tous les procédés existants pour les fonctions de plusieurs variables sont bons, mais tous calculent des minima ou des maxima locaux. Ils sont fondés sur l'hypothèse de l'unicité de l'extremum au voisinage des valeurs initiales injectées dans le calcul.
On s'efforce, en informatique scientifique, de démontrer cette unicité et, lorsqu'on n'y parvient pas, de définir les conditions, les intervalles où elle existe pour s'y placer. C'est généralement un travail difficile, long et pénible. De plus, cette étude préalable garantit qu'il existe un extremum dans certains intervalles, mais non que ce sera l'extremum absolu.

Il existe des méthodes probabilistes de recherche des extrema absolus mais, par nature, elles peuvent seulement garantir que le point trouvé finalement est bien le minimum ou le maximum absolu avec une certaine probabilité, par exemple de 95 %. Elles ne donnent pas de certitude.

Dans les problèmes pratiques utilisant des méthodes de minimisation / maximisation, seul l'extremum absolu peut fournir le meilleur résultat, la meilleure approximation, d'où le grand intérêt d'une méthode déterministe d'obtention de cette valeur. Cet intérêt suppose toutefois que la méthode retenue ne consomme pas trop de temps-machine; celui-ci doit rester comparable avec le temps total nécessaire à l'obtention d'un bon extremum par les méthodes habituelles (durée de l'étude préalable donnant les conditions d'existence et d'unicité, plus durée de rédaction et de frappe du programme, plus temps d'exécution proprement dit).

On sait parfaitement, et depuis longtemps, trouver l'extremum absolu d'une fonction d'une seule variable, y compris quand cette fonction n'est pas exprimable mathématiquement, quand il s'agit d'une sous-routine (on utilise dans ce cas des procédés de parcours et discrétisation).
Les auteurs ont donc imaginé une transformation mathématique qui ramène toute fonction de plusieurs variables à une fonction d'une seule variable avec l'approximation voulue et, en pratique, avec une excellente approximation.
Cette transformation est baptisée Aliénor.

Le principe fondamental d'Aliénor est une propriété particulière aux spirales d'Archimède (voir figure 1). La courbe, d'équation polaire R=T/C, passe à la distance maximum 2π /C de tout point M du plan, cette distance étant mesurée sur le rayon vecteur et T étant positif.
Si le signe de T est quelconque, on obtient la spirale double de la figure 2a, qui passe à la distance maximum π/C de tout point du plan. Les spirales d'Archimède passent par l'origine et leurs équations cartésiennes paramétriques sont :

- X1 = T*COS(T)/C
- X2 = T*SIN(T)/C

On peut donc déjà, par la transformation ci-dessus (baptisons-la 1), ramener toute fonction de deux variables à une fonction de la seule variable T avec l'approximation que l'on veut, c'est-à-dire définir une courbe qui passera à une distance maximum fixée de tout point de la surface représentée par la fonction de deux variables, sous la seule condition que cette fonction soit continue.

Plus précisément, soit Z=f(X1,X2) la fonction examinée: elle représente une surface. Soient Y1 et Y2 les images de X1 et X2 par la transformation (1), c'est-à-dire le point de la spirale d'Archimède d'équation (1) qui est le plus proche de X1-X2. Si la fonction f(X1,X2) est continue, f(Y1,Y2) sera une bonne approximation de f(X1,X2), et même d'autant meilleure que Y1-Y2 sera plus près de X1-X2.
Or :

- la distance ((X1−Y1)^2) + ((X2−Y2)^2) est bornée et reste inférieure à ((2π/C)^2)
- en tout extremum à dérivées partielles nulles, la fonction examinée est continue

Le minimum ou le maximum absolus de l'image Z=f(Y1,Y2) seront donc des valeurs approchées des extrema absolus de l'original Z=f(X1,X2).

En général, si l'on peut définir un parcours suffisamment dense de l'espace des variables, la transformation définie par les équations de ce parcours fournira une valeur approchée des extrema de la fonction examinée.
Il n'est pas difficile de densifier le parcours d'Archimède du plan, ce qui permet de limiter le paramètre C et le pas d'exploration. Il suffit pour cela de connaître les bornes inférieures et supérieures des variables qu'on cherche.
En pratique, cette condition est toujours remplie. Elle peut nécessiter une étude préalable du phénomène traité, mais cette étude est de toute façon indispensable et profitable.

Généralisation

La figure 2 représente une transformation Aliénor à 4 variables et la figure 3 une transformation à 8 variables. Ces figures montrent comment on généralise à un nombre quelconque N de variables. La transformation a une structure d'arbre.
Ainsi, la perte d'information reste bornée et, d'autre part, il est facile d'établir un programme de test qui calculera la distance réduite maximum à laquelle le parcours passe de tout point M de l'espace à N dimensions en fonction de ses paramètres (diviseur C, valeur maximum TM de la variable T d'Aliénor, pas d'exploration P et, on le verra un peu plus loin, facteur de densification K).

Pour N variables transformées, le nombre de niveaux S est la partie entière par excès du logarithme base 2 de N. Il est donc agréablement borné :

- N=2 : S=1
- N=3 ou 4 : S=2
- N=5, 6, 7 ou 8 : S=3
- N=9 à 16, S=4 ...

L'informatique est une science expérimentale et l'expérience nous a appris que :

- Une bonne valeur de TM est (C^S).
- 0,2 est toujours une bonne valeur de pas.
- 2 et 3 sont toujours de bonnes va-

leurs du facteur de densification K.

- De bonnes valeurs du diviseur C sont : 25 pour 2 variables transformées, 7 pour 3 ou 4, 6 pour 5 à 8.

En pratique, les micro-ordinateurs actuels ne peuvent aller au-delà de 16 variables, car le nombre de pas, et donc le temps de calcul, deviennent ensuite prohibitifs. En effet, 6^4 = 1296 et 1296/0,2 = 6480 pas pour 16 variables transformées, soit autant d'exécution des calculs, ce qui représente facilement toute une nuit.

Densification. Obtention du minimum absolu de l'image

L'expérience a appris que la meilleure technique de densification, la plus rapide, employait la fonction INT.
Soient m et M les bornes inférieures et supérieures de la variable X, et Y=f(T) la transformée archimédienne de X, le principe de la densification est :

- Si 0<Y<1 : XA = m + (M−m)*Y
- Autrement : XA = m + (M−m) * (K*Y − INT(K*Y))

XA est l'image de X par Aliénor.

La figure 4 montre l'obtention du minimum absolu de l'image. On prend d'abord un minimum au hasard MR, puis on progresse sur le parcours à partir de T=0 en cherchant des minima successifs à des cotes de plus en plus basses. Le plus petit minimum trouvé est l'approximation cherchée du minimum absolu de la fonction originale de N variables.

Le pas initial dans la sous-routine de minimisation sera 0,1. Il devra être diminué jusqu'à 0,002, ce qui fait 7 passes au facteur de réduction 0,55. Ce dernier est généralement conseillé.
Ensuite, on obtiendra le minimum absolu de l'original par une sous-routine de variations locales, qui n'aura que peu d'itérations à faire.

Formules de transformation

1) Pour 3 variables :

- T1=T*COS(T)/7
- T2=T*SIN(T)/7 et
- X1=T1*COS(T1)/7
- X2=T2*COS(T2)/7
- X3=T2*SIN(T2)/7

2) Pour 4 variables :

- T1, T2, X1, X2 et X3 comme ci-dessus
- X4=T1*SIN(T1)/7

3) Pour 5 variables :

- T1=T*COS(T)/6
- T2=T*SIN(T)/6
- T3=T1*COS(T1)/6
- T4=T1*SIN(T1)/6
- T5=T2*COS(T2)/6
- X1=T3*COS(T3)/6
- X2=T3*SIN(T3)/6
- X3=T4*COS(T4)/6
- X4=T4*SIN(T4)/6
- X5=T5*COS(T5)/6

4) Pour 6 variables, ajouter X6=T5*SIN(T5)/6

5) Pour 7 variables, ajouter T6=T2*SIN(T2)/6 et X7=T6*COS(T6)/6

6) Pour 8 variables, ajouter X8=T6*SIN(T6)/6 ...

Exemple de programme

Le programme ci-joint identifie un développement exponentiel à 4 termes, qui simule l'une des fonctions d'un système compartimental et est censé représenter une concentration.

Le programme principal commence par la fabrication du problème (lignes 2 à 32). On donne en ligne 6 les dates T(I) des mesures. On prend en ligne 10, au hasard, les exposants L(I) de façon vraisemblable, puis de même en ligne 12 les coefficients A(I). Les lignes 14 à 24 constituent la boucle de calcul des valeurs des concentrations C(I) pour les T(I). Ces valeurs sont astreintes par la ligne 22 à être toutes positives, puis subissent une perturbation aléatoire de plus ou moins 0,05*RND(8) pour être rendues vraisemblables.

Sous-routine 400 :transformation Aliénor.

Lignes 420 à 500 :calcul des exposants et des coefficients, calcul des "valeurs calculées" CC(I) de la concentration, calcul de la fonction à minimiser DE qui est 100 fois l'erreur type des moindres carrés de CC(I) exprimée avec 5 décimales. Par souci de réalisme, C(I) et CC(I) sont exprimés avec 3 décimales.

Sous-routine 300 :obtention des minima locaux d'Aliénor, la sous-routine 200 donnant le minimum absolu de la fonction DE des 8 variables L(I) et A(I).

Lignes 34 à 42 :obtention au hasard du point le plus bas parmi 15 (DA, TA), puis, via 300, du minimum correspondant (DR, TR).

Lignes 44 à 58 :exécution de la figure 4, les minima d'Aliénor successifs, de plus en plus bas, sont appelés DZ-TZ. Le dernier trouvé est le plus petit et donc le minimum absolu d'Aliénor; on l'appelle DL et les valeurs correspondantes des L(I) et A(I) sont dites LL(I) et AL(I).

Ligne 60 :prévoit un aléa : le minimum absolu d'Aliénor peut-être DR. De même, la ligne 56 prévoit que la vallée a pu être trouvée en TA-DA par l'aval.

Ligne 62 :appeler 300 au lieu de 400 ne peut qu'améliorer le résultat. Les résultats d'Aliénor sont imprimés aux lignes 68 à 82.

La ligne 86 appelle 200 pour les résultats définitifs qui sont DM (100 fois l'erreur type), les exposants L(I), les coefficients A(I) et les "valeurs calculées" CC(I). Ces résultats sont imprimés aux lignes 88 à 100.

La sous-routine 400 comprend les expressions des L(I) à l'aide de la variable T d'Aliénor (lignes 400 à 418), les initialisations (420 à 426), le calcul des matrices primitives des moindres carrés MC(I,J) et CM(I,J) et des matrices principales M(I,K) et V(I) (lignes 434 à 446), le calcul des A(I) par la méthode de Gauss (lignes 448 à 486), et enfin le calcul des CC(I) puis de DE (lignes 488 à 499).

La sous-routine de minima d'Aliénor (300 à 332) est du type "à balancement", minimisation quadratique, tri et reprise avec réduction du pas. Le pas initial est défini en 300 : c'est le pas d'exploration (0,2) multiplié par DA/100 pour qu'il y ait très peu de reprises. Rappelons en effet que la première valeur de DA est le plus petit des 15 nombres DE pris au hasard et que les valeurs suivantes sont toutes inférieures à DR, et de plus en plus petites. On se procure d'abord 5 valeurs encadrantes, les X et les Y, on sort la plus petite (lignes 316 à 322), et on reprend avec elle et un pas réduit. Les Y sont les DE et les X sont les angles. Le résultat final est baptisé TM-DM.

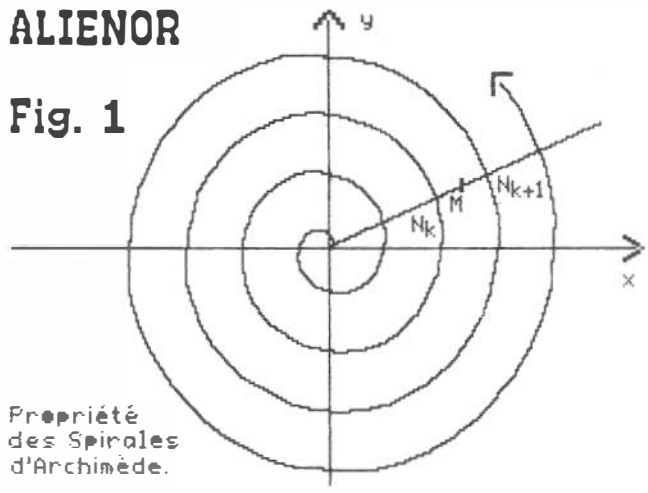
La sous-routine 200 est organisée de la même façon et agit sur les 4 exposants, qui y prennent les appellations Y, les DE correspondants étant appelés X.

Les résultats d'Aliénor sont excellents : l'approximation trouvée (2.48515 %) est inférieure au quantum de la perturbation aléatoire (5 %). Il ne faut pas croire que c'est une exception ; c'est la **règle**. Tous ceux qui utiliseront ce programme sur leur Apple pourront s'en convaincre.
Les résultats définitifs sont encore un peu meilleurs, mais à peine. Pourtant, la sous-routine 200 procure une modification significative des exposants L(I) et des coefficients A(I).

■

ALIENOR

Fig. 1



Propriété des Spirales d'Archimède.

Fig. 2a

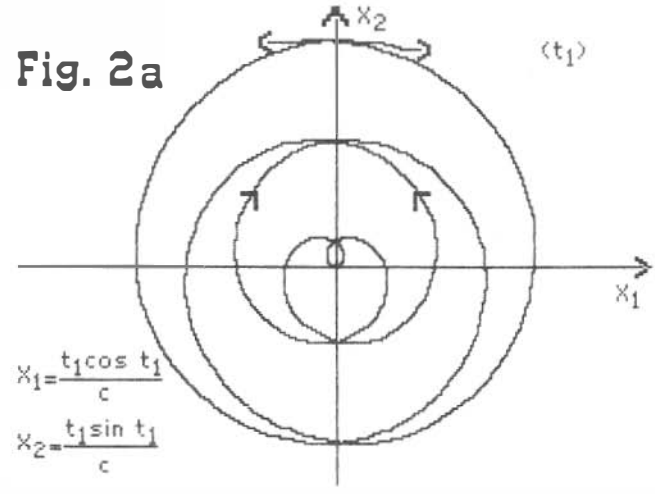


Fig. 2b

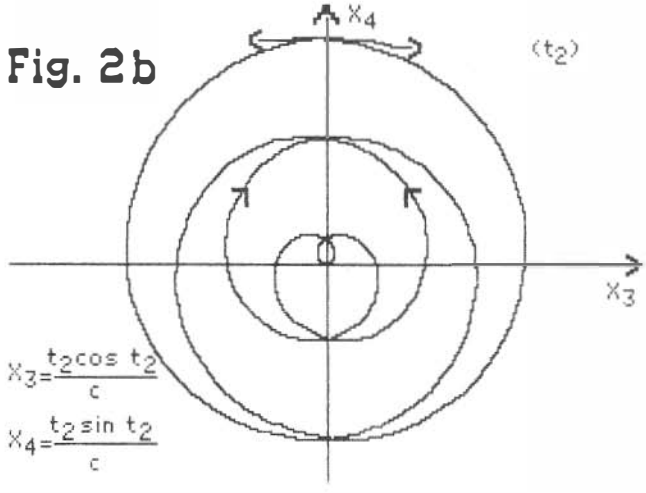
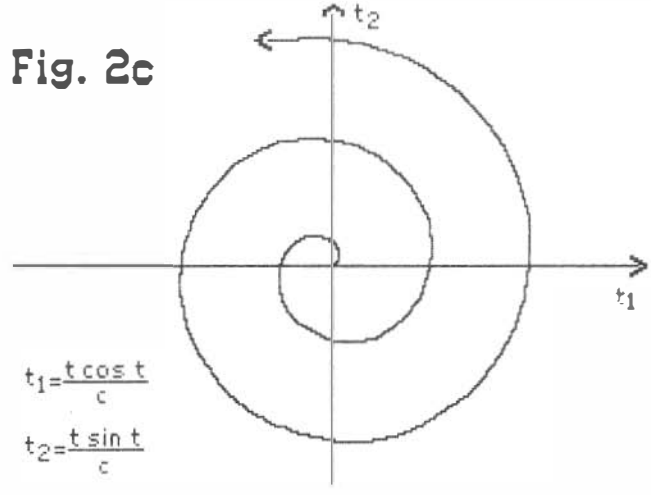
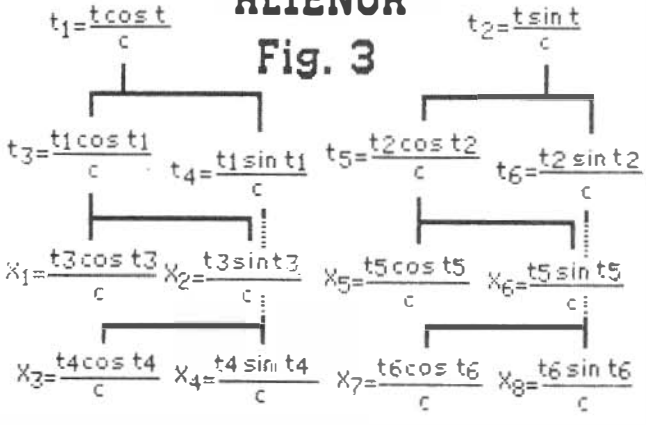


Fig. 2c



ALIENOR

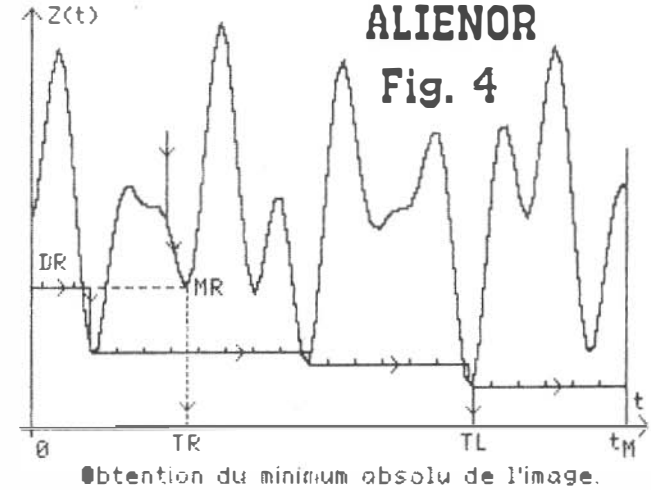
Fig. 3



Structure d'arbre

ALIENOR

Fig. 4



Obtention du minimum absolu de l'image.

Programme ALIENOR

```
1   REM  ** ALIENOR **

2   DIM T(11),C(11),CC(11),MC(11,3)
      ,CM(3,11),M(3,3),V(3),L(3),LL
      (3),A(3),AL(3)

4   DIM LX(3),P(3),X(5),Y(5),G(3)
6   DATA .5,1,2,4,6,8,11,15,18,21,2
      4,30
8   FOR I = 0 TO 11: READ T(I): NEXT

10 L(0) = .30 *  RND (1):L(1) =  RND
      (1):L(2) = L(1) + (1 - L(1)) *
```

```
     RND (1):L(3) = 4 + 2 *  RND
    (1)
12 A(0) = 2 + 4 *  RND (1):A(1) =
     - 2 - 2 *  RND (I):A(2) = .5
     - 5 *  RND (1):A(3) = 3 + 5 *
     RND (1)
14 I = 0:C(I) = 0
16 J = 0:SI =  INT ( RND (1) * 100
    )
18 C(I) = C(I) +  EXP ( - T(I) * L
    (J)) * A(J)
20  IF J <  = 2 THEN J = J + 1: GOTO
    18
22  IF C(I) <  = 0 THEN 12
23 C(I) = C(I) * (1 + .05 *  RND (
    1) * ( - 1) ^ SI):C(I) = .001
     *  INT (C(I) * 1000 + .5)
24  IF I <  = 10 THEN I = I + 1: GOTO
    16
25  PRINT  CHR$ (4)"PR#1"
27  PRINT "DONNEES"
28  FOR I = 0 TO 11 STEP 3
29  PRINT "C("I")="C(I)"      C("I +
    1")="C(I + 1)"      C("I + 2")
    ="C(I + 2)
30  NEXT I
31  PRINT "**********************
    *******"
32  PRINT  CHR$ (4)"PR#0"
34 T = 50 *  RND (1): GOSUB 400:DA
     = DE:TA = T
36  FOR II = 0 TO 15:T = 50 *  RND
    (1): GOSUB 400:DI = DE
38  IF DI <  = (DA) THEN DA = DI:T
    A = T
40  NEXT II
42  GOSUB 300:DR = DM:TR = TM: PRINT
    "DR="DR
44 TZ = 0:DZ = DR
46 Q = 0
48 T = TZ + .2 * Q: GOSUB 400:DA =
    DE:TA = T
50  IF T >  = 50 THEN 60
52  IF DA > DZ THEN Q = Q + 1: GOTO
    48
54 GOSUB 300:DZ = DM:TZ = TM:TC =
    TM
56  IF TZ <  = (TA + .2) THEN TZ =
    TA + .2
58  GOTO 46
60  IF TZ = 0 THEN TC = TR
62 TA = TC:DA = DZ: GOSUB 300:DL =
    DM
64  PRINT  CHR$ (4)"PR#1"
68  PRINT "RESULTATS D'ALIENOR"
70  FOR II = 0 TO 3:LL(II) = L(II)
    :AL(II) = A(II)
72  PRINT "LL("II")="LL(II)"      A
    L("II")="AL(II)
74  NEXT I1
75  FOR M = 0 TO I1 STEP 3
76  PRINT "CC("M")="CC(M)"      CC(
    "M + 1")="CC(M + 1)"      CC("
    M + 2")="CC(M + 2)
77  NEXT M
78  PRINT "DL="DL
80  PRINT "**********************
    ***"
82  PRINT  CHR$ (4)"PR#0"
86  GOSUB 200
88  PRINT  CHR$ (4)"PR#I"
92  PRINT "RESULTATS DEFINITIFS"
94  PRINT "DM="DM
96  FOR II = 0 TO 3: PRINT "L("II"
    )="L(II)"      A("II")="A(II):
    NEXT II
97  FOR II = 0 TO 11 STEP 3
98  PRINT "CC("II")="CC(II)"      C
    C("II + I")="CC(II + 1)"
    CC("II + 2")="CC(II + 2)
99  NEXT II
100  PRINT  CHR$ (4)"PR#0"
104  END
200 RG = 0:DX = DL
202  FOR I2 = 0 TO 3:LX(I2) = LL(I
    2):P(I2) = LL(I2) * DL / 100:
    NEXT I2
204  FOR I2 = 0 TO 3:Y(0) = LX(I2)
    :X(0) = DX
206 Y(1) = Y(0) + P(I2):L(I2) = Y(
    1): GOSUB 420:X(1) = DE
208  IF X(1) < X(0) THEN Y(0) = Y(
    1):X(0) = X(1): GOTO 206
210 Y(2) = Y(0):X(2) = X(0)
212 Y(3) = Y(2) - P(I2):L(I2) = Y(
    3): GOSUB 420:X(3) = DE
214  IF X(3) < X(2) THEN Y(2) = Y(
    3):X(2) = X(3): GOTO 212
216 Y(4) = Y(0) + P(I2) * X(0) / (
    X(0) + X(1)):L(I2) = Y(4): GOSUB
    420:X(4) = DE
218 Y(5) = Y(2) - P(I2) * X(2) / (
    X(2) + X(3)):L(I2) = Y(5): GOSUB
    420:X(5) = DE
220 DX = X(0):LX(I2) = Y(0)
222  FOR J2 = 1 TO 5
224  IF X(J2) <  = X(0) THEN DX =
    X(J2):LX(I2) = Y(J2)
226  NEXT J2
228  NEXT I2
230  IF (.55 ^ RG) <  = .01 THEN 2
    36
232  FOR I2 = 0 TO 3:P(I2) = .55 *
    P(I2): NEXT I2
234 RG = RG + 1: GOTO 204
236 L(0) = LX(0):L(1) = LX(1):L(2)
     = LX(2):L(3) = LX(3)
238  GOSUB 420:DM = DE
240  RETURN
300 P = DA * .002:X(0) = TA:Y(0) =
    DA:RF = 0
302 X(I) = X(0) + P:T = X(1): GOSUB
    400:Y(1) = DE
304  IF Y(1) <  = (Y(0) - 1E - 5) THEN
    X(0) = X(I):Y(0) = Y(1): GOTO
    302
306 X(2) = X(0):Y(2) = Y(0)
308 X(3) = X(2) - P:T = X(3): GOSUB
    400:Y(3) = DE
310  IF Y(3) <  = (Y(2) - 1E - 5) THEN
    X(2) = X(3):Y(2) = Y(3): GOTO
    308
312 X(4) = X(0) + P * Y(0) / (Y(0)
     + Y(1)):T = X(4): GOSUB 400:
```

```
     Y(4) = DE
314 X(5) = X(2) - P * Y(2) / (Y(3)
     + Y(2)):T = X(5): GOSUB 400:
     Y(5) = DE
316 TM = X(0):DM = Y(0)
318  FOR I1 = 1 TO 5
320  IF Y(I1) < Y(0) THEN TM = X(I
     1):DM = Y(I1)
322  NEXT I1
324  IF P <  = .002 THEN 330
326 X(0) = TM:Y(0) = DM:P = .55 *
     P
328 RF = RF + 1: GOTO 302
330 T = TM: GOSUB 400
332  RETURN
400 T0 = T *  COS (T) / 7:T1 = T *
     SIN (T) / 7
402 T2 = T0 *  COS (T0) / 7:T3 = T
     0 *  SIN (T0) / 7:T4 = T1 *  COS
     (T1) / 7:T5 = T1 *  SIN (T1) /
     7
404  IF T2 >  = 0 AND T2 <  = .3 THEN
     L(0) = T2: GOTO 408
406 L(0) = T2 - .3 *  INT (3.3333 *
     T2)
408  IF T3 >  = 0 AND T3 <  = .333
     4 THEN L(1) = 3 * T3: GOTO 41
     2
410 L(1) = 3 * T3 -  INT (3 * T3)
412  IF T4 >  = 0 AND T4 <  = .333
     4 THEN L(2) = L(1) + (1 - L(1
     )) * 3 * T4: GOTO 416
414 L(2) = L(1) + (1 - L(1)) * (3 *
     T4 -  INT (3 * T4))
416  IF T5 >  = 0 AND T5 <  = .333
     4 THEN L(3) = 6 * T5 + 4: GOTO
     420
418 L(3) = 4 + 6 * T5 - 2 *  INT (
     3 * T5)
420 DE = 0:SE = 0
422  FOR I = 0 TO 3:V(I) = 0:M(I,0
     ) = 0:M(I,1) = 0:M(I,2) = 0:M
     (I,3) = 0
424 L(I) =  ABS (L(I)):L(I) =  INT
     (L(I) * 1E4 + .5) * 1E - 4
426  NEXT I
434  FOR I = 0 TO 11: FOR J = 0 TO
     3
436 E =  EXP ( - T(I) * L(J)):MC(I
     ,J) = E:CM(J,I) = E
438  NEXT J,I
440  FOR I = 0 TO 3: FOR J = 0 TO
     11
442 V(I) = V(I) + CM(I,J) * C(J)
444  FOR K = 0 TO 3:M(I,K) = M(I,K
     ) + CM(I,J) * MC(J,K): NEXT K

446  NEXT J,I
448  FOR H = 3 TO 1 STEP  - 1: FOR
     I = H - 1 TO 0 STEP  - 1
450  IF M(H,H) <  > 0 THEN 470
452  FOR L = H - 1 TO 0 STEP  - 1
454  IF M(L,H) <  > 0 THEN 460
455  NEXT L
456 DE = 10000: GOTO 500
460  REM ALIENOR
462  FOR W = 0 TO 3
464 G(W) = M(L,W):M(L,W) = M(H,W):
     M(H,W) = G(W)
466  NEXT W
468 V1 = V(L):V(L) = V(H):V(H) = V
     1
470 V(I) = V(I) - M(I,H) * V(H) /
     M(H,H)
472  FOR J = 0 TO H
474 M(I,J) = M(I,J) - M(I,H) * M(H
     ,J) / M(H,H)
476  NEXT J,I,H
478 A(0) = V(0) / M(0,0)
479 A(0) =  INT (A(0) * 1E4 + .5) *
     1E - 4
480  FOR I = 1 TO 3:S = 0
482  FOR K = 0 TO I - 1:S = S + M(
     I,K) * A(K): NEXT K
484 A(I) = (V(I) - S) / M(1,I)
485 A(I) =  INT (A(I) * 1E4 + .5) *
     1E - 4
486  NEXT 1
488  FOR 1 = 0 TO 11:CC(I) = 0
490  FOR J = 0 TO 3:CC(I) = CC(I) +
     A(J) * MC(I,J): NEXT J
492 CC(I) =  INT (CC(I) * 1000 + .
     5) * .001
494 SE = SE +  ABS (CC(I)):DE = DE
     + (CC(I) - C(I)) ^ 2
496  NEXT I
498 DE = 100 *  SQR (12 * DE) / SE

499 DE =  INT (DE * 1E5 + .5) * 1E
     - 5
500  RETURN
```

```
1  REM  ** PROGRAMME DE TEST **
                    (ALIENOR)


2  INPUT C:TM = C ^ 2 + 1
4  DIM DL(20):CR = 0
6 L(0) = .3 *  RND (1):L(1) =  RND
     (1):L(2) = L(I) + (1 - L(1)) *
     RND (1):L(3) = 4 + 2 *  RND
     (1)
8 TA = TM *  RND (1):T = TA: GOSUB
     400:DA = DE
10  FOR II = 0 TO 15:T = TM *  RND
     (1): GOSUB 400:DI = DE
12  IF DI < (DA) THEN TA = T:DA =
     DE
14  NEXT II
16  GOSUB 300:DR = DL:TR = TL: PRINT
     "DR="DR
18 OZ = DR:TZ = 0
20 Q = 0
22 TA = TZ + .2 * Q:T = TA: GOSUB
     400:DA = DE
24  IF T >  = TM THEN 34
26  IF DA > DZ THEN Q = Q + 1: GOTO
     22
28  GOSUB 300:DZ = DL:TZ = TL:TC =
     TL
30  IF TZ <  = (TA + .2) THEN TZ =
     TA + .2
```

```
32  GOTO 20
34  IF TZ = 0 THEN TC = TR
36 TA = TC:DA = DZ: GOSUB 300:DL(C
     R) = DL: PRINT "DL("CR")="DL(
     CR)
38  IF CR >  = 20 THEN 42
40 CR = CR + 1: GOTO 6
42 DS = DL(0):DI = DL(0):DL = DL(0
     ) ^ 2
44  FOR J = 1 TO 20
46  IF DL(J) ) DL(0) THEN DS = DL(
     J)
48  IF DL(J) < DL(0) THEN DI = DL(
     J)
50 DL = DL + DL(J) ^ 2
52  NEXT J
54 DL =  SQR (DL / 21)
56  PRINT  CHR$ (4)"PR#1"
60  PRINT "RESULTATS DU TEST"
62  PRINT "DL="DL"%    DS="DS"%
      DI="DI"%    C="C
64  PRINT  CHR$ (4)"PR#0"
70  END
300 P = DA * .002:X(0) = TA:Y(0) =
     DA:RF = 0
302 X(1) = X(0) + P:T = X(1): GOSUB
     400:Y(1) = DE
304  IF Y(1) <  = (Y(0) - 1E - 5) THEN
     X(0) = X(1):Y(0) = Y(1): GOTO
     302
306 X(2) = X(0):Y(2) = Y(0)
308 X(3) = X(2) - P:T = X(3): GOSUB
     400:Y(3) = DE
310  IF Y(3) <  = (Y(2) - 1E - 5) THEN
     X(2) = X(3):Y(2) = Y(3): GOTO
     308
312 X(4) = X(0) + P * Y(0) / (Y(0)
      + Y(1)):T = X(4): GOSUB 400:
     Y(4) = DE
314 X(5) = X(2) - P * Y(2) / (Y(3)
      + Y(2)):T = X(5): GOSUB 400:
     Y(5) = DE
316 TL = X(0):DL = Y(0)
318  FOR I1 = 1 TO 5
320  IF Y(I1) < Y(0) THEN TL = X(I
     1):DL = Y(I1)
322  NEXT I1
324  IF P <  = .002 THEN 330
326 X(0) = TL:Y(0) = DL:P = .55 *
     P
328 RF = RF + 1: GOTO 302
330 T = TL: GOSUB 400
332  RETURN
400 T0 = T *  COS (T) / C:T1 = T *
      SIN (T) / C
402 T2 = T0 *  COS (T0) / C:T3 = T
     0 *  SIN (T0) / C:T4 = T1 *  COS
     (T1) / C:T5 = T1 *  SIN (T1) /
     C
404  IF T2 >  = 0 AND T2 <  = .3 THEN
     M(0) = T2: GOTO 408
406 M(0) = T2 - .3 *  INT (3.3333 *
     T2)
408  IF T3 >  = 0 AND T3 <  = .333
     4 THEN M(1) = 3 * T3: GOTO 41
     2
410 M(1) = 3 * T3 -  INT (3 * T3)
412  IF T4 >  = 0 AND T4 <  = .333
     4 THEN M(2) = M(1) + (1 - M(1
     )) * 3 * T4: GOTO 416
414 M(2) = M(1) + (1 - M(1)) * (3 *
     T4 -  INT (3 * T4))
416  IF T5 >  = 0 AND T5 <  = .333
     4 THEN M(3) = 6 * T5 + 4: GOTO
     420
418 M(3) = 4 + 6 * T5 - 2 *  INT (
     3 * T5)
420 DE = 0:SE = 0
422  FOR I = 0 TO 3:SE = SE + M(I)
     :DE = DE + (M(I) - L(I)) ^ 2:
      NEXT I
424 DE = 200 *  SQR (DE) / SE
426 DE = 1E - 5 *  INT (DE * 1E5 +
     .5)
430  RETURN
```

Un menu à la carte

Michel Crimont

Il s'agit d'une UNIT avec une fonction principale renvoyant un caractère pris dans un menu. L'affichage se fait horizontalement ou verticalement, le choix se fait soit en tapant la première lettre de l'option, soit en positionnant la zone inversée avec les flèches et en faisant RETURN.

Cette UNIT utilise elle-même l'UNIT LIB contenant PINFO et PCHAR (cf numéros précédents).

L'utilisation se fait par exemple en déclarant :
VAR M : TABMEN;
puis dans le programme :

```
PROCEDURE          INITM(VAR
M1:TABMEN);
BEGIN
ZEROMENU(M1);
M1!0! := 'TITRE'; M1!1! = 'Lire';
M1!2! := 'Changer';
etc....
END
```

Le programme principal commencera par INITM(M); on utilisera la fonction par :

```
CASE MENU(F,M) OF
'L': LIRE;
'C': CHANGER;
```

La présentation à l'écran dépend du paramètre F :

Si F=0, seule la ligne d'affichage est effacée; sinon, tout l'écran est effacé avant affichage.
Si F<=0, il y a affichage du menu sur une ligne.
Si F>0, il y a affichage vertical, le chiffre F représentant l'espacement entre les lignes. F=1 : toutes les lignes. F=2 : toutes les deux lignes.
A noter que la ligne 0 du tableau sert de titre au menu et n'est pas utilisée comme option.

```
     (*$S+*)
UNIT GETMENU;
          INTERFACE
USES LIB;(* Pour PCOM=PRENCAR sans affichage, et PRINFO avec HT:=CHR(21);
          INV:=CHR(15);NORM:=CHR(14);LF:=CHR(10);VT:= CHR(11) pour APPLE IIe*)
TYPE TABMEN = PACKED ARRAY[0..10] OF STRING[20];(*La ligne 0 sert au titrage *)
  FUNCTION MENU(F:INTEGER; VAR M : TABMEN):CHAR;
  PROCEDURE ZEROMENU(VAR M : TABMEN);
          IMPLEMENTATION
FUNCTION MENU;
VAR NUMOPTION,I,MAXOP,MX:INTEGER;  OPTION:CHAR;  TOTCOM,DIR:SET OF CHAR;

  PROCEDURE PRINTLIGNE(I:INTEGER;S:STRING;GR:CHAR);
  BEGIN
    IF F <=0 THEN
    BEGIN
      GOTOXY(I*13,0);WRITE(GR,' ',S[1],'(',COPY(S,2,LENGTH(S)-1),',')
    END ELSE
    BEGIN
      GOTOXY(20,2+F*I);WRITE(GR,' ',S[1],' - ',S);
    END;
    WRITE(NORM);
  END;

BEGIN
  TOTCOM:=[ESC];  NUMOPTION:=1;  MAXOP:=0;  GOTOXY(0,0);
  IF F<>0 THEN WRITE(EFE) ELSE WRITE(EFL);
  IF F>0 THEN GOTOXY((79-LENGTH(M[0])) DIV 2,0);  WRITE(M[0]);
  IF F>0 THEN BEGIN DIR:=[CR,LF,VT];MX:=10 END
          ELSE BEGIN DIR:=[CR,BS,HT];MX:=5 END; (* touches de direction*)
  FOR I:=1 TO MX DO
    IF M[I]<>'' THEN
    BEGIN
      MAXOP:=MAXOP+1;PRINTLIGNE(I,M[I],NORM);TOTCOM:=TOTCOM+[M[I,1]];
      (*totcom contient les premi(res lettres des options*)
    END;
  PRINTLIGNE(1,M[1],INV);
  REPEAT
    OPTION:=PCOM(TOTCOM+DIR);
    IF OPTION IN TOTCOM THEN BEGIN MENU:=OPTION;EXIT(MENU) END;(* lettre *)
    PRINTLIGNE(NUMOPTION,M[NUMOPTION],NORM);(* dir r{{crire en normal *)
    CASE ORD(OPTION) OF
      8,21,10,11:BEGIN
                    (*deplacement*)
                    IF OPTION IN [BS,VT] THEN NUMOPTION:=NUMOPTION-1
                                         ELSE NUMOPTION:=NUMOPTION+1;
                    IF NUMOPTION<1 THEN NUMOPTION:=MAXOP;(*circulaire*)
                    IF NUMOPTION>MAXOP THEN NUMOPTION:=1;
                    PRINTLIGNE(NUMOPTION,M[NUMOPTION],INV);(*inv la ligne*)
                  END;
      13     :BEGIN MENU:=M[NUMOPTION,1];EXIT(MENU) END;(*opt sur return*)
    END;
  UNTIL OPTION=ESC;(* ESC pour sortir du menu *)
END;

PROCEDURE ZEROMENU;
BEGIN   FILLCHAR(M,SIZEOF(M),CHR(0))  END;

BEGIN
END.
```

Des listages propres ...

Pour lister un programme sur imprimante sans voir "]LIST" sur la feuille ni devoir faire "PR#0" après, il faut faire "PR#1 : LIST".

Pour comprendre cela, lire le "Trucs et astuces" du numéro 7 en page 26. Malheureusement, ":PR#1 : CATALOG" ne marche pas !

Une routine de tri rapide en assembleur

Gérard Rigaud

Le programme présenté ici permet de trier des tableaux de tout type, à une ou deux dimensions, avec association éventuelle d'un deuxième tableau contenant des informations parallèles.

Pour les tableaux à deux dimensions, la première représente les "rubriques" et la seconde les "articles". Le nombre de rubriques ne doit pas dépasser 51, mais le nombre d'articles ne dépend que de la place mémoire disponible.

L'appel de la routine se fait au moyen de l'ampersand (&); la syntaxe en est la suivante :

&T,I,A(I,0),A(I,N)

avec :

- A : nom du tableau (par exemple : A, B%, ZZ$...)
- I : indice sur lequel on trie (numéro de la rubrique)
- A(I,0) : premier article du tri
- A(I,N) : dernier article du tri.

On peut ne trier qu'une partie des articles en fixant les bornes. Par exemple :

&T,I,A(I,3),A(I,40)

Si le tableau n'a qu'une dimension, I doit être à zéro. La syntaxe devient alors :

&T,0,A(0),A(N)

On peut associer un autre tableau au tableau trié, avec une syntaxe du type :

&T,I,A$(I,2),A$(I,50),A(0,2)

où A(0,2) désigne le premier élément trié du tableau associé à A$, le tri lui-même se faisant sur la rubrique I de A$, du 2éme au 50ème article.

Il est également possible de trier sur une partie d'article, avec une instruction du type :

&S,8,I,A$(I,2),A$(I,50)

par laquelle le tri se ferait à partir du 8ème caractère des articles de A$.
La profondeur admise ici pour le point de départ de la partie d'article retenue est limitée à 15 caractères (ligne 123 du programme source), mais il est possible de la modifier avant assemblage ou par un POKE à partir du BASIC.

Les petits programmes de démonstration listés ci-après vous permettront de bien saisir les modalités d'utilisation de cette routine.

■

Programme TRITABLEAU DOS TOOL KIT

```
 1 ;
 2 FLAG    EQU  $6        ;Flag r(e),entier
 3 IND     EQU  $9        ;Indice du tri
 4 ECART   EQU  $7        ;Ecart entre articles
 5 ECO     EQU  $8        ;Ecart indice 0 et tri
 6 START   EQU  $19       ;Pointeur 1er (l(ment du tri
 7 FIN     EQU  $18       ;Pointeur dernier (l(ment
 8 ART1    EQU  $FA       ;Pointeur temporaire
 9 ART2    EQU  $FC       ;Pointeur temporaire
10 CHA2    EQU  $FE       ;Pointeur temporaire
11 MOVE    EQU  $1E       ;Pointeur temporaire
12 CSTART  EQU  $E0       ;Pointeur 1er (l(ment C0tableau
13 CART1   EQU  $E2       ;Pointeur C0tableau
14 CART2   EQU  $E4       ;Pointeur C0tableau
15 TECART  EQU  $10       ;Ecart 1er tableau
16 TECO    EQU  $E6       ;Ecart (l(ment 1er tableau
17 CECART  EQU  $E7       ;Ecart C0tableau
18 PARTIEL EQU  $E8
19 SARA    EQU  $98       ;Pointeur du tableau
20 CHRGET  EQU  $81
21 CHRGOT  EQU  $87
22 FRMNUM  EQU  $D067     ;Evalue dans FAC
23 CHKCOM  EQU  $DEBE     ;Teste ','
24 MOVFM   EQU  $EAF9     ;Transfert M vers FAC
25 SOUST   EQU  $EBB2     ;FAC - ARG
26 GETADR  EQU  $E752     ;Rend entier FAC
27 PTRGET  EQU  $DFE3
28         ORG  $9AA6
29 ;------Initialisation AMPERSAND------
30         LDA  #>ST
31         STA  $3F6
32         LDA  #<ST
33         STA  $3F7
34         RTS
35 ;--------------------D(but du programme-----------
36 ST      CMP  #$54      ;'T'
37         BEQ  TRI
38         CMP  #$53      ;'S'
39         BEQ  SORT
40         JMP  $DEC9     ;SYNTAX ERROR
41 SORT    JSR  CHRGET
42         JSR  CHKCOM
43         JSR  FRMNUM
44         JSR  GETADR
45         DEY
46         DEY
47         STY  PARTIEL
48         JMP  TS
49 TRI     LDA  #$FF
50         STA  PARTIEL
51         JSR  CHRGET
52 TS      JSR  CHKCOM
53         JSR  FRMNUM
54         JSR  GETADR    ;Rend FAC entier
55         STY  IND       ;Indice du tri
56         JSR  CHKCOM
57         JSR  PTRGET
58         STA  START
59         STA  ART1
60         STY  START+1
61         STY  ART1+1
```

```
 62          JSR  CHKCOM
 63          JSR  PTRGET
 64          STA  FIN
 65          STY  FIN+1
 66          JSR  CALCUL     ;Initialise pointeurs
 67          LDA  ECO
 68          STA  TECO
 69          LDA  ECART
 70          STA  TECART
 71          STX  FLAG
 72          JSR  CHRGOT
 73          CMP  #$2C       ;COTABLEAU ?
 74          BNE  DEBUT      ;Non
 75          JSR  CHRGET
 76          JSR  PTRGET
 77          STA  CSTART
 78          STA  CART1
 79          STY  CSTART+1
 80          STY  CART1+1
 81          JSR  CALCUL
 82          LDA  ECART
 83          STA  CECART
 84 ;------Début du tri------
 85 DEBUT    LDA  #0
 86          STA  IND        ;Flag d'échange
 87 SUIVANT  CLC              ;Calcul adresse 2ème article
 88          LDA  TECART
 89          ADC  ART1
 90          STA  ART2
 91          LDA  #0
 92          ADC  ART1+1
 93          STA  ART2+1
 94          CLC              ;Calcul adresse 2ème article
 95          LDA  CECART
 96          ADC  CART1
 97          STA  CART2
 98          LDA  #0
 99          ADC  CART1+1
100          STA  CART2+1
101          LDA  FLAG
102          CMP  #5         ;Réel ?
103          BEQ  REEL       ;Oui
104          TAY
105          DEY
106          CMP  #2         ;Entier ?
107          BEQ  ENTIER     ;OUI
108          LDA  (ART1),Y
109          STA  MOVE+1
110          LDA  (ART2),Y
111          STA  CHA2+1
112          DEY
113          LDA  (ART1),Y
114          STA  MOVE
115          LDA  (ART2),Y
116          STA  CHA2
117          LDY  PARTIEL
118 COMP     INY
119          LDA  (CHA2),Y
120          CMP  (MOVE),Y
121          BCC  DEPL       ;CHA2 < CHA1
122          BNE  RIEN       ;CHA2 > CHA1
123          CPY  #15        ;Profondeur du tri
124          BCC  COMP
125          BCS  RIEN
126 ENTIER   SEC
127          LDA  (ART2),Y
128          SBC  (ART1),Y
129          DEY
130          LDA  (ART2),Y
131          SBC  (ART1),Y
132          BPL  RIEN       ;ART2 = ou > ART1
133          BMI  DEPL       ;ART2 < ART1
134 ;------Recommence à l'article 1------
135 BOUCLE   LDA  CSTART     ;1er article
136          STA  CART1
137          LDA  CSTART+1
138          STA  CART1+1
139          LDA  START      ;1er article
140          STA  ART1
141          LDA  START+1
142          STA  ART1+1
143          BNE  DEBUT
144 ;------Article suivant------
145 SUITE    LDA  CART2      ;Article suivant
146          STA  CART1
147          LDA  CART2+1
148          STA  CART1+1
149          LDA  ART2       ;Article suivant
150          STA  ART1
151          LDA  ART2+1
152          STA  ART1+1
153          BNE  SUIVANT
154 ;------Test de fin du tri------
155 RIEN     LDA  ART2
156          CMP  FIN
157          BNE  SUITE
158          LDA  ART2+1
159          CMP  FIN+1
160          BNE  SUITE
161          LDA  IND
162          BNE  BOUCLE
163          RTS
164 ;------------------------------------------------
165 REEL     LDA  ART2
166          LDY  ART2+1
167          JSR  MOVFM      ;ART2 dans FAC
168          LDA  ART1
169          LDY  ART1+1
170          JSR  SOUST      ;ART2 - ART1
171          BPL  RIEN       ;ART2 = ou > ART1
172 ;------Déplacement des pointeurs------
173 DEPL     SEC              ;Calcul élément 0 de ART1
174          LDA  ART1
175          SBC  TECO
176          STA  ART1
177          LDA  ART1+1
178          SBC  #0
179          STA  ART1+1
```

```
180          SEC              ;Calcul (l(ment 0 de ART2
181          LDA  ART2
182          SBC  TEC0
183          STA  MOVE
184          LDA  ART2+1
185          SBC  #0
186          STA  MOVE+1
187          LDY  TECART
188          DEY
189          STA  IND         ;Positionne flag
190 SWAP     LDA  (ART1),Y
191          PHA
192          LDA  (MOVE),Y
193          STA  (ART1),Y    ;(change 2 dans 1
194          PLA
195          STA  (MOVE),Y    ;(change 1 dans 2
196          DEY
197          BPL  SWAP
198          LDY  CECART
199          BEQ  RIEN
200          DEY
201 SWAP1    LDA  (CART1),Y
202          PHA
203          LDA  (CART2),Y
204          STA  (CART1),Y ;(change 2 dans 1
205          PLA
206          STA  (CART2),Y ;(change 1 dans 2
207          DEY
208          BPL  SWAP1
209          BMI  RIEN
210 ;-------Sous programme recherche pointeurs-------
211 CALCUL   LDX  #$3
212          BIT  $11         ;Chaine ?
213          BMI  DIM1        ;Oui
214          LDX  #$5
215          BIT  $81         ;R(el ?
216          BPL  DIM1        ;Oui
217          LDX  #$2
218 DIM1     LDY  #$4         ;Dimension ?
219          LDA  (SARA),Y
220          CMP  #$1         ;Une seule ?
221          BEQ  D1          ;Oui
222          LDY  #$8
223          LDA  (SARA),Y    ;Taille 1)re dimension
224 D1       TAY
225          LDA  #0
226          STA  ECART
227          STA  EC0
228          STA  CECART
229          CLC              ;Calcul (cart entre articles
230 D2       TXA              ;5 . 3 . 2
231          ADC  ECART
232          STA  ECART
233          DEY
234          BNE  D2
235          LDY  IND
236          BEQ  D4
237          CLC              ;(cart (l(ment 0 et tri
238 D3       TXA              ;5 . 3 . 2
239          ADC  EC0
240          STA  EC0
241          DEY
242          BNE  D3
243 D4       RTS
```

RECAPITULATION

```
9AA6- A9 B1
9AA8- 8D F6 03 A9 9A 8D F7 03
9A80- 60 C9 54 F0 1A C9 53 F0
9A88- 03 4C C9 DE 20 B1 00 20
9AC0- BE DE 20 67 DD 20 52 E7
9AC8- 88 88 84 E8 4C D6 9A A9
9AD0- FF 85 E8 20 B1 00 20 BE
9AD8- DE 20 67 DD 20 52 E7 84
9AE0- 09 20 BE DE 20 E3 DF 85
9AE8- 19 85 FA 84 1A 84 FB 20
9AF0- BE DE 20 E3 DF 85 1B 84
9AF8- 1C 20 01 9C A5 08 85 E6
9B00- A5 07 85 1D 86 06 20 B7
9B08- 00 C9 2C D0 15 20 B1 00
9B10- 20 E3 DF 85 E0 85 E2 84
9B18- E1 84 E3 20 01 9C A5 07
9B20- 85 E7 A9 00 85 09 18 A5
9B28- 1D 65 FA 85 FC A9 00 65
9B30- FB 85 FD 18 A5 E7 65 E2
9B38- 85 E4 A9 00 65 E3 85 E5
9B40- A5 06 C9 05 F0 6B A8 88
9B48- C9 02 F0 22 B1 FA 85 1F
9B50- B1 FC 85 FF 88 B1 FA 85
9B58- 1E B1 FC 85 FE A4 E8 C8
9B60- B1 FE D1 1E 90 5B D0 38
9B68- C0 0F 90 F3 B0 32 38 B1
9B70- FC F1 FA 88 B1 FC F1 FA
9B78- 10 26 30 45 A5 E0 85 E2
9B80- A5 E1 85 E3 A5 19 85 FA
9B88- A5 1A 85 FB D0 94 A5 E4
9B90- 85 E2 A5 E5 85 E3 A5 FC
9B98- 85 FA A5 FD B5 FB D0 86
9BA0- A5 FC C5 1B D0 E8 A5 FD
9BA8- C5 1C D0 E2 A5 09 D0 CC
9BB0- 60 A5 FC A4 FD 20 F9 EA
9BB8- A5 FA A4 FB 20 B2 EB 10
9BC0- DF 38 A5 FA E5 E6 85 FA
9BC8- A5 FB E9 00 85 FB 38 A5
9BD0- FC E5 E6 85 1E A5 FD E9
9BD8- 00 85 1F A4 1D 88 85 09
9BE0- B1 FA 48 B1 1E 91 FA 68
9BE8- 91 1E 88 10 F3 A4 E7 F0
9BF0- AF 88 B1 E2 48 B1 E4 91
9BF8- E2 68 91 E4 88 10 F3 30
9C00- 9F A2 03 24 11 30 08 A2
9C08- 05 24 81 10 02 A2 02 A0
9C10- 04 B1 9B C9 01 F0 04 A0
9C18- 08 B1 9B A8 A9 00 85 07
9C20- 85 08 85 E7 18 8A 65 07
9C28- 85 07 88 D0 F8 A4 09 F0
9C30- 09 18 8A 65 08 85 08 88
9C38- D0 F8 60
```

EXEMPLES

```
10  PRINT  CHR$ (4)"BRUNTRITABLEAU
     .OBJ0"
20  TEXT : CLEAR : HOME
30  DIM A$(3,9),A(9): PRINT "TRI S
     UR 3 eme ELEMENT"
40  FOR I = 0 TO 9: FOR J = 0 TO 3
     :A$(J,I) =  CHR$ (74 - I): PRINT
     A$(J,I)".";:A(I) = I: PRINT A
     (I)" ";: NEXT : PRINT : NEXT
     : VTAB 1: HTAB 25: PRINT "TRI
      MACHINE"
50  & T,3,A$(3,0),A$(3,9)
60  FOR I = 0 TO 9: HTAB 24: FOR J
      = 0 TO 3: PRINT A$(J,I)"."A(
     I)" ";: NEXT : PRINT : NEXT
70  PRINT "TAPER UNE TOUCHE";: GET
     X$: HTAB 1: INVERSE : PRINT "
     LE MEME EXEMPLE AVEC COTABLEA
     U": NORMAL
75  CLEAR : DIM A$(3,9),A(9): PRINT
     "TRI SUR 3 eme ELEMENT"
80  FOR I = 0 TO 9: FOR J = 0 TO 3
     :A$(J,I) =  CHR$ (74 - I): PRINT
     A$(J,I)".";:A(I) = I: PRINT A
     (I)" ";: NEXT : PRINT : NEXT
     : VTAB 13: HTAB 23: PRINT "MA
     CHINE COTABLEAU"
90  & T,3,A$(3,0),A$(3,9),A(0): REM
        Transfere aussi A() qui pe
     ut avoir egalement 2 dimensio
     ns
100  FOR I = 0 TO 9: HTAB 24: FOR
     J = 0 TO 3: PRINT A$(J,I)"."A
     (I)" ";: NEXT : PRINT : NEXT

110  VTAB 23: END
```

```
5  PRINT  CHR$ (4)"BRUNTRITABLEAU.
     OBJ0"
10  TEXT : CLEAR : HOME
20  DIM A$(20): PRINT "TABLEAU"
30  FOR I = 0 TO 20:A$(I) =  CHR$
     (85 - I) +  CHR$ (B5 - I) +  STR$
     (65 + I): PRINT A$(I): NEXT :
      VTAB 1: HTAB 9: PRINT "TRINO
     RMAL"
40  & T,C,A$(0),A$(20): REM   TRI
     TABLEAU ENTIER
50  FOR I = 0 TO 20: HTAB 10: PRINT
     A$(I): NEXT : VTAB 1: HTAB 19
     : PRINT "TRI 3eme"
60  & S,3,0,A$(0),A$(20): REM TRI
     A PARTIR DU 3eme CARACTERE
70  FOR I = 0 TO 20: HTAB 20: PRINT
     A$(I): NEXT : VTAB 1: HTAB 29
     : PRINT "TRI 4eme"
80  & S,4,0,A$(0),A$(20): REM  TRI
      A PARTIR DU 4eme CARACTERE
90  FOR I = 0 TO 20: HTAB 30: PRINT
     A$(I): NEXT : INVERSE : PRINT
     "TRI A PARTIR DU 1,3,4 CARACT
     ERE": NORMAL
100  VTAB 23: END
```

```
5  PRINT  CHR$ (4)"BRUNTRITABLEAU.
     OBJ0"
10  TEXT : CLEAR : HOME
20  DIM A%(50): PRINT "TRI TABLEAU
      ENTIER 1 DIMENSION"
30  FOR I = 0 TO 50:A%(I) = 100 -
     I: PRINT A%(I);"   ";: NEXT :
      PRINT : PRINT "TRI MACHINE"
40  & T,0,A%(0),A%(50): REM   TRI
     TABLEAU ENTIER
50  FOR I = 0 TO 50: PRINT A%(I);"
        ";: NEXT : PRINT
60  PRINT "TAPER UNE TOUCHE";: GET
     X$: PRINT : CLEAR
70  DIM A%(50): PRINT "TRI TABLEAU
      ENTIER 1 DIMENSION"
80  FOR I = 0 TO 50:A%(I) = 100 -
     I: PRINT A%(I);"   ";: NEXT :
      PRINT : PRINT "LE MEME , TRI
      BASIC , C'EST PLUS LONG !"
90 X = 0: FOR I = 50 TO 1 STEP  -
     1: IF A%(I) >  = A%(I - 1) THEN
     110
100 X = A%(I):A%(I) = A%(I - 1):A%
     (I - 1) = X
110  NEXT : IF X THEN 90
120  FOR I = 0 TO 50: PRINT A%(I);
     "   ";: NEXT : PRINT
```

```
5  PRINT  CHR$ (4)"BRUNTRITABLEAU.
     OBJ0"
10  TEXT : CLEAR : HOME
20  DIM A%(3,10): PRINT "TRI TABLE
     AU ENTIER"
30  FOR I = 0 TO 10: FOR J = 0 TO
     3:A%(J,I) = 20 - I + J: PRINT
     A%(J,I);"  ";: NEXT : PRINT "
     *";: NEXT : PRINT : PRINT "TR
     I MACHINE"
40  & T,0,A%(0,0),A%(0,10): REM  T
     RI SUR ELEMENT 0 TABLEAU ENTI
     ER
50  FOR I = 0 TO 10: FOR J = 0 TO
     3: PRINT A%(J,I);"  ";: NEXT
     : PRINT "*";: NEXT : PRINT
60  PRINT "TAPER UNE TOUCHE";: GET
     X$: PRINT : CLEAR
70  DIM A%(3,10): PRINT "TRI TABLE
     AU ENTIER"
80  FOR I = 0 TO 10: FOR J = 0 TO
     3:A%(J,I) = 20 - I + J: PRINT
     A%(J,I);"  ";: NEXT : PRINT "
     *";: NEXT : PRINT : PRINT "LE
      MEME , TRI BASIC"
90 X = 0: FOR I = 10 TO 1 STEP  -
     1: IF A%(0,I) >  = A%(0,I - I
     ) THEN 110
100  FOR X = 0 TO 3:Z = A%(X,I):A%
     (X,I) = A%(X,I - I):A%(X,I -
     1) = Z: NEXT
110  NEXT : IF X THEN 90
120  FOR I = 0 TO 10: FOR J = 0 TO
     3: PRINT A%(J,I);"  ";: NEXT
     : PRINT "*";: NEXT : PRINT
```

```
5  PRINT  CHR$ (4)"BRUNTRITABLEAU.
     OBJ0"
10  TEXT : CLEAR : HOME
20  DIM A$(50): PRINT "TRI TABLEAU
     CHAINE 1 DIMENSION"
30  FOR I = 0 TO 50:A$(I) =  STR$
     (99 - I): PRINT A$(I);"   ";:
     NEXT : PRINT : PRINT "TRI MA
     CHINE"
40  & T,0,A$(0),A$(50): REM  TRI T
     ABLEAU ENTIER
50  FOR I = 0 TO 50: PRINT A$(I);"
       ";: NEXT : PRINT
60  PRINT "TAPER UNE TOUCHE";: GET
     X$: PRINT : CLEAR
70  DIM A$(50): PRINT "TRI TABLEAU
     CHAINE 1 DIMENSION"
80  FOR I = 0 TO 50:A$(I) =  STR$
     (99 - I): PRINT A$(I);"   ";:
     NEXT : PRINT : PRINT "LE MEM
     E , TRI BASIC , C'EST PLUS LO
     NG !"
90 X = 0: FOR I = 50 TO 1 STEP  -
     1: IF A$(I) >  = A$(I - 1) THEN
     110
100 X$ = A$(I):A$(I) = A$(I - 1):A
     $(I - 1) = X$:X = 1
110  NEXT : IF X THEN 90
120  FOR I = 0 TO 50: PRINT A$(I);
     "   ";: NEXT : PRINT
```

```
5  PRINT  CHR$ (4)"BRUNTRITABLEAU.
     OBJ0"
10  TEXT : CLEAR : HOME
20  DIM A$(3,10): PRINT "TRI TABLE
     AU CHAINE"
30  FOR I = 0 TO 10: FOR J = 0 TO
     3:A$(J,I) =  STR$ (20 - 1 + J
     ): PRINT A$(J,I);"  ";: NEXT
     : PRINT "*";: NEXT : PRINT : PRINT
     "TRI MACHINE"
40  & T,0,A$(0,0),A$(0,10): REM TR
     I SUR ELEMENT 0 TABLEAU ENTIE
     R
50  FOR I = 0 TO 10: FOR J = 0 TO
     3: PRINT A$(J,I);"  ";: NEXT
     : PRINT "*";: NEXT : PRINT
60  PRINT "TAPER UNE TOUCHE";: GET
     X$: PRINT : CLEAR
70  DIM A$(3,10): PRINT "TRI TABLE
     AU CHAINE"
80  FOR I = 0 TO 10: FOR J = 0 TO
     3:A$(J,I) =  STR$ (20 - 1 + J
     ): PRINT A$(J,I);"  ";: NEXT
     : PRINT "*";: NEXT : PRINT : PRINT
     "LE MEME , TRI BASIC"
90 X = 0: FOR I = 10 TO 1 STEP  -
     1: IF A$(0,I) >  = A$(0,I - 1
     ) THEN 110
100  FOR X = 0 TO 3:Z$ = A$(X,I):A
     $(X,I) = A$(X,I - I):A$(X,I -
     1) = Z$: NEXT :X = 1
110  NEXT : IF X THEN 90
120  FOR I = 0 TO 10: FOR J = 0 TO
     3: PRINT A$(J,I);"  ";: NEXT
     : PRINT "*";: NEXT : PRINT
```

```
5  PRINT  CHR$ (4)"BRUNTRITABLEAU.
     OBJ0"
10  TEXT : CLEAR : HOME
20  DIM A(50): PRINT "TRI TABLEAU
     REEL 1 DIMENSION"
30  FOR I = 0 TO 50:A(I) = 100 - I
     : PRINT A(I);"   ";: NEXT : PRINT
     : PRINT "TRI MACHINE"
40  & T,0,A(0),A(50): REM  TRI TA
     BLEAU ENTIER
50  FOR I = 0 TO 50: PRINT A(I);"
       ";: NEXT : PRINT
60  PRINT "TAPER UNE TOUCHE";: GET
     X$: PRINT : CLEAR
70  DIM A(50): PRINT "TRI TABLEAU
     REEL 1 DIMENSION"
80  FOR I = 0 TO 50:A(I) = 100 - I
     : PRINT A(I);"   ";: NEXT : PRINT
     : PRINT "LE MEME , TRI BASIC
     , C'EST PLUS LONG !"
90 X = 0: FOR I = 50 TO 1 STEP  -
     1: IF A(I) >  = A(I - 1) THEN
     110
100 X = A(I):A(I) = A(I - 1):A(I -
     1) = X
110  NEXT : IF X THEN 90
120  FOR I = 0 TO 50: PRINT A(I);"
       ";: NEXT : PRINT
```

```
5  PRINT  CHR$ (4)"BRUNTRITABLEAU.
     OBJ0"
10  TEXT : CLEAR : HOME
20  DIM A(3,10): PRINT "TRI TABLEA
     U REEL"
30  FOR I = 0 TO 10: FOR J = 0 TO
     3:A(J,I) = 20 - I + J: PRINT
     A(J,I);"  ";: NEXT : PRINT "*
     ";: NEXT : PRINT : PRINT "TRI
     MACHINE"
40  & T,0,A(0,0),A(0,10): REM  TRI
     SUR ELEMENT 0 TABLEAU ENTIER

50  FOR I = 0 TO 10: FOR J = 0 TO
     3: PRINT A(J,I);"  ";: NEXT :
     PRINT "*";: NEXT : PRINT
60  PRINT "TAPER UNE TOUCHE";: GET
     X$: PRINT : CLEAR
70  DIM A(3,10): PRINT "TRI TABLEA
     U REEL"
80  FOR I = 0 TO 10: FOR J = 0 TO
     3:A(J,I) = 20 - 1 + J: PRINT
     A(J,I);"  ";: NEXT : PRINT "*
     ";: NEXT : PRINT : PRINT "LE
     MEME , TRI BASIC"
90 X = 0: FOR I = 10 TO 1 STEP  -
     1: IF A(0,I) >  = A(0,I - 1) THEN
     110
100  FOR X = 0 TO 3:Z = A(X,I):A(X
     ,I) = A(X,I - 1):A(X,I - 1) =
     Z: NEXT
110  NEXT : IF X THEN 90
120  FOR I = 0 TO 10: FOR J = 0 TO
     3: PRINT A(J,I);"  ";: NEXT :
     PRINT "*";: NEXT : PRINT
```

Effets stroboscopiques sur la page graphique

Olivier Herz

L'idée de ce programme m'est venue d'une idée originale publiée dans Softside Magazine.

Ceux qui ont acheté la disquette du numéro 6 de Pom's ont sans doute remarqué un curieux effet stroboscopique dans le programme de démonstration de la Tortue Ampersand de Jacques Duma (DEMO TORTUE). Cet effet se présentait sous la forme de bandes en inverse se déplaçant lentement sur une image positive. Il avait lieu alors que l'on "switchait" entre les pages graphiques haute résolution. la page 2 étant le négatif de la page 1: il était dû à un phénomène stroboscopique entre la fréquence de balayage de l'écran vidéo et la fréquence de switch entre les pages graphiques. Si par exemple l'on se contente d'exécuter les instructions en assembleur suivantes, la période (2 switches) est de 11 micro-secondes (voir "La Pratique de l'Apple II", volume 3) :

```
ZER BIT $C054
BIT $C055
JMP ZER
```

En faisant varier cette fréquence, on peut obtenir des effets stroboscopiques.

Nous vous proposons de bien visualiser un tel effet en faisant varier la fréquence de switch des pages graphiques avec les paddles : la manette 0 contrôlera le temps d'affichage de la page 1 et la manette 1 celui de la page 2. On peut utiliser au choix un joystick ou 2 manettes.

Nous utiliserons les deux routines (relogeables) en assembleur suivantes, entrées en mémoire par la méthode de SIMLAM (voir Pom's 2 page 61 ou Recueil page 94).

Inversion de la page 1 avec résultat dans la page 2 :

```
MEM1 EQU 252
MEM2 EQU 254
A0 20 LDY #32 :octet haut de la page 1
84 FD STY MEM1+1
A0 40 LDY #64 :octet haut de la page 2
84 FF STY MEM2+1
A0 00 LDY #0 :pour l'adressage indirect indexé
84 FC STY MEM1
84 FE STY MEM2
ZER B1 FC LDA (MEM1).Y :octet de la page 1
49 FF EOR #255 :on l'inverse
91 FE STA (MEM2).Y .résultat dans la page 2
C8 INY :on boucle sur Y de 0 à 255 (1 page mémoire)
D0 F7 BNE ZER
E6 FD INY MEM1+1
E6 FF INY MEM2+1
A5 FD LDA MEM1+1
C9 40 CMP #64 :on boucle sur les pages 32 à 63
D0 ED BNE ZER
60 RTS
```

Switch des pages avec pauses proportionnelles aux paddles :

```
PREAD EQU $FB1E :lit dans A la manette X
KBD EQU $C000 :clavier (-16384)
KBDSTRB EQU $C010 :strobe du clavier (-16368)
PAGE1 EQU $C054 .switche vers la page 1 (-16300)
PAGE2 EQU $C055 .switche vers la page 2 (-16299)

ONE 2C 54 C0 BIT PAGE1 .switche vers la page 1
A2 00 LDX #0
20 1E FB JSR PREAD :lit la manette 0
2C 55 C0 BIT PAGE2 .switche vers la page 2
A2 01 LDX #1
20 1E FB JSR PREAD :lit la manette 1
2C 00 C0 BIT KBD :touche enfoncée ?
10 EB BPL ONE
2C 10 C0 BIT KBDSTRB .remise à 0 du clavier
60 RTS
```

Bien sûr, la 2éme page graphique peut ne pas être l'inverse de la 1ère. De plus, cet effet reste si on remplace l'une ou les deux pages graphiques haute résolution par une page texte ou graphique basse résolution, car seule la vitesse de balayage de l'écran entre en jeu, pas le contenu de l'écran. Bien entendu, dans ces différents cas, il faudra utiliser les autres switches graphiques de l'Apple. ■

```
10  REM  STROBOSCOPE
20  REM  ==========

100  TEXT : HOME : HGR : VTAB 21: INPUT
     "NOM DE LA PAGE GRAPHIQUE ? "
     ;A$: POKE  - 16302,0: PRINT  CHR$
     (4)"BLOAD"A$",A$2000"
110 A$ = "300:2C 54 C0 A2 00 20 1E
      FB 2C 55 C0 A2 01 20 1E FB 2
     C 00 C0 10 EB 2C 10 C0 60": GOSUB
     1000
120 A$ = "320:A0 20 84 FD A0 40 84
      FF A0 00 84 FC 84 FE B1 FC 4
     9 FF 91 FE C8 D0 F7 E6 FD E6
     FF A5 FD C9 40 D0 ED 60": GOSUB
     1000
130  CALL 800: CALL 768: TEXT : HOME
     : END
1000 A$ = A$ + " N D9C6G": FOR I =
     1 TO  LEN (A$): POKE 511 + I,
      ASC ( MID$ (A$,I,1)) + 128: NEXT
     : POKE 72,0: CALL  - 144: RETURN
```

MacArticle

Daniel Garric

Je n'y croyais pas. Mon métier, c'est d'écrire. Trente années de clavier intensif mal tempéré, de la machine mécanique à l'Apple II, avec une impossibilité radicale de rédiger plus de deux lignes - illisibles - à la main. Alors la souris de Macintosh, non merci, ce n'était pas pour moi.

Deux heures ont suffi pour que je me retrouve, inconditionnel, dans le camp de la souris, avec des crampes à la main droite le premier jour et un immense sentiment de liberté les jours suivants.
Je répète : mon métier c'est d'écrire, pas de faire des routines, encore moins de la routine. Et surtout pas de taper, même si je tape très vite. Il peut m'arriver de faire des programmes en BASIC, en Pascal, en Superpilot. Ce n'est pas l'essentiel.

Alors j'ai vite compris que Macintosh était la première "machine" qui permette de créer, sans aucune contrainte externe, à un détail près que je soulignerai tout à l'heure. Je dis bien "détail" parce que c'est passager, et j'écris bien "machine" parce que le mot "ordinateur" ne me vient pas au clavier. J'ai envie de dire "robot de bureau".

Premier temps, je sors Mac de son carton. Emerge un petit bonhomme ventru haut comme trois pommes (bien sûr), avec un petit écran et un petit clavier détachable. Attention, encore un cliché à voiler une fois pour toutes : tous les claviers alphanumériques à touches classiques sont de la même taille. Donc, celui de Mac a beau paraître petit, il ne l'est pas, pour la bonne raison qu'il n'y a pas de "petit clavier". Pourquoi semble-t-il "petit" ? Parce qu'il ne lui reste que deux touches de fonction, dont l'une, au moins, le célèbre "Control" qui s'orne ici d'un sigle bizarre que l'on identifie très vite, ne demeure que par concession au passé.

Une fois posé le bonhomme sur le bureau - qu'il a l'air menu à côté du "gros" Apple //e - j'enfonce la fiche dans la "machine" puis dans la prise; je branche le clavier. En une dizaine de secondes, c'est fini, c'est-à-dire que tout commence. J'enfonce dans la fente avant un gros bout de carton avec une espèce de serrure en métal que je sors de ma poche (il paraît que c'est une disquette 3,5 pouces), et je mets le contact. L'écran vit, des carrés, des rectangles gris sillonnent l'espace gris-blanc. Un menu ? Il paraît.

Moi, je vois une autre chose, étrange, inattendue : un manuscrit copié par un moine du Moyen-Age. Cette première impression se confirmera les jours suivants : avec Mac, nous franchissons une formidable étape de rétro-innovation, c'est-à-dire de bond dans les deux directions du temps, le passé et l'avenir.
Le passé : jusqu'à l'apparition de l'imprimerie, il ne pouvait y avoir de texte que personnel (on ne disait pas encore "personnalisé"), puisque chaque copie, dans sa forme, était un original, avec la calligraphie, les ornements, la mise en page du copiste. L'imprimerie, fantastique découverte bien sûr, est venu jeter là-dessus sa chape d'uniformité. Beaucoup de livres pour beaucoup de monde, mais tous les mêmes (les livres) ou presque. Allez vous étonner, après cela, que des Apollinaire, des Michaux ou des Prévert aient éprouvé le besoin de faire des calligrammes, de réinventer la forme, de lui redonner sa place.
L'avenir : c'est exactement ce que ramène Macintosh. Les poètes, les écrivains, les créateurs deviennent tous des éditeurs, des imprimeurs, sur cette "machine" dont l'écran reproduit presque parfaitement la page écrite, la page manuscrite.

Sur ma disquette, deux programmes, MacWrite - traitement de texte - et MacPaint - traitement graphique. Du jamais vu, du jamais imaginé, l'un comme l'autre. MacWrite : neuf jeux de caractères (dont un gothique), six variétés d'édition (ombre, relief, italique ...), cinq dimensions d'impression, du 9 au 24 points, que l'on peut combiner, que l'on peut augmenter en créant ses propres caractères. Et toujours, toujours, toujours sans avoir jamais, jamais, jamais besoin de connaître un seul nom de touche de contrôle, de se demander si c'est CTRL-E(rase) ou CTRL-D(elete) qui signifie "effacer", s'il faut faire ESC-A ou ESC-Q, CTRL-S(ave), CTRL-L(oad).

La libération, la voici : tout est à l'écran, écrit en clair : "fermer", "couper", "ouvrir", "coller", "effacer", "enregistrer" (ouf ! Du même coup, nous voici libérés des affreux barbarismes du genre "sauvegarder", "sauver", comme si tous les textes et programmes d'ordinateurs étaient toujours des chefs d'oeuvre en péril). Et pourtant, votre écran semble vide, en tout six mots sur la ligne du haut. Autant de tiroirs que vous ouvrez d'un coup de souris magique, sans plus y penser, comme si votre main faisait désormais "cerveau à part".

Or, toutes ces commandes en arborescence - il doit y en avoir plus de cinquante dont vous n'avez pas à vous souvenir - demeurent les mêmes pour tous les programmes de Macintosh. J'ai essayé Multiplan et MacPaint (je vais y revenir). Il y aura, très vite, la série des PFS (les Personal Files System, chef d'oeuvre de simplicité et d'efficacité), Multichart, MacDraw et bien d'autres programmes. Je sais déjà que je n'aurai pas à les apprendre, à me redemander dix fois si c'est CTRL-P, CTRL-Q ou même "T" pour "Transfert". Cette fois, nous y sommes : l'ordinateur est au service de l'homme, de tous les hommes qui savent lire et écrire, et non plus de l'informaticien seul. Et si Macintosh était le premier vrai micro-ordinateur ? Il faut lui donner un autre nom, celui-là est déjà pris. Pourquoi pas un BUREAUBOT ?

Je reviens à MacWrite : avec Macintosh, pas question d'écrire en style linéaire. La machine ne l'est pas puisqu'elle permet sans cesse de se promener partout dans le système. Avec MacWrite, je l'avoue, j'ai un truc sentimental tout bête. J'ai deux enfants en bas âge, Nathanaël et Marie-Laëtitia. Ma femme n'est pas très attirée par les ordinateurs et je sais bien pourquoi : ils ne permettent pas d'orthographier correctement le prénom de nos enfants. Quelle "machine" vous sort le "i" avec un tréma et le célèbre "e pris dans a ou dans o" ? O surprise ! O merveille ! Mac sait faire tout ça, et bien d'autres choses encore. J'ai encore le clavier QWERTY et pourtant, tous les accents sont là. A l'écran. Y compris le circonf.exe, qui se pavane sur les "a", les "o", les "u" le plus naturellement du monde, comme s'il était tout à fait courant, pour un accent circonflexe, de se montrer grandeur nature sur un écran d'ordinateur.

Je saute à MacPaint. C'est - évidemment - tout simple : deux coups de souris. Là, ce sera bref. Parce que MacPaint ne se raconte pas, il se dessine. Et je ne sais pas dessiner. Mais je peux tout de même comprendre qu'il y a là tout ce dont un dessinateur a toujours eu besoin et n'a jamais osé espérer trouver un jour : une gomme, du papier, un crayon... Mais aussi des mines de tailles différentes à un coup de souris les unes des autres, des fonds, des tas de fonds que l'on peut répandre à volonté, aux dimensions et aux formes que l'on veut. Et une main qui vous fait faire le tour de la page en promenant votre dessin dans les recoins secrets de l'écran. Et une loupe pour les retouches, et une brosse, et une possibilité de changer la taille des dessins, et un lasso pour capturer la partie du chef d'oeuvre dont vous désirez incruster votre

texte. A moins que vous ne fassiez le contraire, que vous mettiez du texte dans le dessin : MacPaint accepte tous les jeux de caractères de MacWrite, avec même quelques tailles supplémentaires. Je regrette encore plus que jamais de ne pas savoir dessiner. Mais je suis heureux pour mes amis publicitaires, créateurs de mode, futurs auteurs de jeux d'aventure, de programmes didactiques, artistiques de tous poils. La définition, 512 par 342 points, suffit largement pour faire une recherche très élaborée, d'autant plus que vous imprimez...

J'allais oublier l'imprimante, justement, l'Imagewriter (tout est dans son nom), inséparable compagne de Mac : silencieuse, solide, graphique, accessible instantanément. Avec une petite coquetterie : trois modes d'impression en texte. Brouillon : j'appuie et c'est parti. C'est rapide, et c'est vraiment un brouillon que vous pouvez retravailler sur papier. Au niveau supérieur, c'est standard : impression honnête, plus lente. Avec "Qualité Supérieure", c'est tout simplement super : l'imprimante se met à faire des pleins et des déliés comme un "instit" de la vieille école, ceux qui vous tapaient sur les doigts si vous n'écriviez pas en ronde avec une plume sergent-major. Mac est plus gentil, il le fait faire par Imagewriter.

J'en arrive aux deux contraintes. La première : il y a tellement de jeux de caractères, tellement de possibilités que vous ne pouvez jamais savoir combien de caractères, combien de lignes vous avez tapés. Dommage, mais compréhensible. Plus grave : un document ne peut avoir plus de 8 pages. Après, il faut en ouvrir un autre. Et si vous écrivez un roman où les chapitres ont trente ou cinquante pages, que vous voulez couper, insérer de la 3ème à la 24ème ? Il faudra sans doute attendre la fin de l'année, lorsque Mac grossira, qu'il passera de 128 à 512 K de mémoire centrale. En attendant, écrivez plutôt des nouvelles, ou arrangez-vous pour que vos chapitres soient si bien écrits et composés qu'il devienne inutile de les retrafiquer.

Restent les programmes. Oui, vous aurez un BASIC presque Microsoft, avec des lignes pas forcément numérotées : les numéros, c'est pour les "GOTO" et autres anachronismes du même genre. Et un "Instant Pascal", comme il existe aux Etats-Unis un "Instant Coffee" ou café soluble. Le Pascal est soluble, lui, parce qu'il a perdu son compilateur dans l'opération, ou du moins, s'il l'a conservé, vous n'en savez rien du tout. Vous écrivez votre programme sur la gauche de l'écran, un coup de souris magique et il se déroule directement sur la partie droite, vous signale toutes les erreurs, tous les trucs qui ne vont pas. Et vous réparez sur le champ de bataille. Autant dire que s'il est beau, le Pascal, tout comme le BASIC, sa puissance est mesurée. Formidable pour apprendre, mais si vous avez des ambitions de type Bill Gates ou Bill Budge, alors non. Allez écrire vos Multiplan ou autres Pinball Set sur Lisa, un nouveau Lisa compatible Mac (ou l'inverse). Et vous en serez récompensés : une figure animée sur Mac, c'est une gravure d'Holbein en mouvement (si le talent y est, bien sûr).

Manque la couleur, c'est dommage. Mais la couleur y est, camouflée dans les PROMs, prête à sortir lorsqu'un Roland Moréno pas quelconque voudra bien aller l'y débusquer. Le vrai problème, c'est l'écran. La télé n'est pas assez bonne et l'écran est cher. Dans six mois, dans un an, voire deux, lorsque le nombre de Macintosh vendus planera dans les un ou deux millions... ■

Magic Window II à l'essai

Alexandre Avrane

La majorité des logiciels de micro-informatique professionnelle se divisent en trois catégories principales : les éditeurs de texte, les tableurs et les gestionnaires de fichiers. Depuis l'apparition de l'Apple //e et de son clavier amélioré, la compétition entre éditeurs de texte est devenue féroce: récemment arrivé, Magic Window II (MW2) est destiné à remplacer l'ancienne version de Magic Window, dont le chiffre des ventes déclinait dramatiquement face à la poussée des nouveaux éditeurs, Apple Writer //e et WordStar principalement.

Repris par Bill Depew, il s'adresse aux possesseurs d'Apple II Plus (avec extension de 16K au moins) ou de //e. Il conserve les qualités originelles de son prédécesseur en incluant plus de souplesse d'utilisation, mais demeure très conservateur : aucun nouveau concept du traitement de texte n'a été introduit.

Adaptable à la plupart des configurations

Magic Window II se compose de l'habituel classeur renfermant la documentation et deux disquettes protégées. La première permet de sauvegarder, sur une disquette vierge, les paramètres de la configuration de l'utilisateur ; MW2 offre à ce niveau une grande diversité de choix et accepte la plupart des cartes 80 colonnes et interfaces d'imprimantes, ainsi que l'existence d'une éventuelle ROM LC, d'un clavier amélioré, ou de la modification de la touche Shift.

La deuxième disquette est, bien sûr, la plus utilisée du système Magic Window, et contient le programme proprement dit. Après avoir "booté", il faut indiquer si l'on désire le chargement optionnel du module permettant l'affichage sur 70 colonnes, très agréable si l'on ne dispose pas de carte 80 colonnes. Il faut ensuite placer la disquette sur laquelle ont été stockés les paramètres de configuration, ou taper un simple Return pour utiliser les options par défaut de la disquette système. Il s'agit donc d'une procédure relativement longue, mais qui offre l'avantage de charger en une fois tous les modules du programme ; la disquette système peut donc être immédiatement rangée dans son étui.

Les utilisateurs des logiciels d'ARTSCI, et en particulier de Magicalc (cf. Pom's 9), ne seront pas dépaysés, car les menus sont très similaires et se distinguent par une excellente ergonomie. Un bon point donc, reflétant la philosophie générale de Magic Window II, qui se veut directement utilisable par tout opérateur non-informaticien.

Une documentation facilement assimilable

Cette approche se retrouve également dans la documentation américaine, qui tente en premier lieu de dédramatiser l'apprentissage des relations homme-ordinateur pour un "non-initié". L'approche pédagogique est très progressive et le lecteur est invité à travailler sur plusieurs exercices. En revanche, la documentation se révèle quasiment muette sur les aspects techniques: je n'ai pu déterminer, par exemple, si Magic Window II utilisait automatiquement une éventuelle carte d'extension mémoire supplémentaire comme Magicalc, autre produit d'ARTSCI, qui gère jusqu'à 512K de RAM. La traduction en français, réalisée par Ordinateur Express, reprend la documentation originale, malheureusement en plus succint : pas d'exercices proposés. Il y manque également l'index et quelques détails (touches à utiliser pour générer des caractères spéciaux).

L'utilisation des deux manuels sera donc nécessaire.

Une utilisation très compartimentée par des menus

Six sections principales sont offertes à partir du menu :

- l'éditeur de texte proprement dit:
- le sous-système de gestion des fichiers:
- le sous-système de formatage permet de spécifier les marges d'une page de texte:
- le sous-système d'impression:
- le sous-système de configuration. enfin, permet de modifier la configuration du clavier. de l'affichage vidéo ou des slots sans devoir recharger le programme.

Remarquons qu'il n'existe pas de fonction d'aide (help) disponible directement.

Visualisation du texte page par page

L'éditeur de texte est. bien évidemment. constamment utilisé. On est tout de suite frappé par l'ordonnancement ergonomique des touches de fonction : il ne m'a fallu qu'une dizaine de minutes pour les mémoriser. alors que je "sèche" encore parfois sur d'autres éditeurs que j'utilise couramment ! Un autre point que j'estime très positif est l'absence de décalage d'un caractère. sur la ligne sur laquelle se trouve le curseur. comme on peut l'expérimenter sur Apple Writer où le curseur est. par défaut. en mode d'insertion.

Un texte correspond à un ensemble maximum de 160*112 caractères par page et l'écran vidéo correspond à une fenêtre de 40. 70 ou 80 colonnes sur 22 lignes. Cette fenêtre se déplace automatiquement pour suivre le curseur et l'on peut donc taper un texte au kilomètre sans regarder l'écran.

Pour obtenir un affichage sur 70 colonnes en haute résolution. on doit charger le module adéquat au moment du boot. il ne faut cependant pas en abuser car la mémoire disponible est alors réduite de 26 à 13K.

Une particularité de Magic Window II est l'affichage permanent des coupures de pages. ce qui permet de monter ou abaisser de quelques lignes un paragraphe qui ne doit pas être tronqué. et évite donc de pénibles essais à tâtons avec l'imprimante. Il est de plus possible de spécifier une ligne de titre répétée à chaque page. et d'écrire une ligne indifféremment en justification à gauche. à droite. centrée ou totale.

Un texte facile à entrer, difficile à corriger

Les options de correction de l'éditeur ne sont malheureusement guère pratiques : il n'est pas possible de supprimer ou déplacer un gros bloc de texte. et les fonctions de recherche ou remplacement d'une chaîne de caractères sont rudimentaires : impossible de spécifier une page ou une colonne pour la recherche. de convertir en majuscules une chaîne en minuscules et vice-versa. de rechercher des caractères spéciaux ou de contrôle.

Magic Window II offre donc des qualités indéniables d'esthétique et d'ergonomie pour la saisie rapide d'un texte. mais il sera parfois nécessaire de transférer le fichier vers un autre éditeur pour une correction complexe.

Une impression facile

Le sous-système d'impression offre la saisie de nombreux paramètres d'édition : nombre d'exemplaires. numérotation automatique des pages. etc. L'impression est. de plus. facilitée par la visualisation permanente du texte tel qu'il sera imprimé.

Une option intéressante est la possibilité de demander l'impression d'une liste de fichiers contenus sur une disquette dans l'ordre désiré et de manière automatique: ce n'est certes pas aussi performant que ce qu'il est possible d'obtenir avec le langage WPL d'Apple Writer //e. mais offre une manière simple et élégante d'obtenir un résultat proche.

Il est également possible de demander une impression vers le lecteur de disquette pour une retransmission ultérieure par modem ou un autre périphérique.

Une gestion de fichier parfois étonnante

Tout d'abord. oh surprise. MW2 gère ses fichiers avec ce monstre préhistorique de lenteur qu'est le DOS 3.3 ! En corollaire. il est impossible de transférer des fichiers Pascal. CP/M (et. bien sûr. ProDOS). Magic Window impose le suffixe ".MW" à ses fichiers. Pourquoi une telle limitation qui complique les transferts entre éditeurs ?

L'utilisateur peut sauvegarder ou charger son texte par un fichier binaire. lisible uniquement par Magic Window II. qui inclut le texte lui-même ainsi que les paramètres de tabulation et de formatage de ce texte. Excellente idée. puisqu'elle évite d'utiliser un fichier supplémentaire de paramètres. mais pourquoi ne pas l'avoir approfondie. par exemple en compactant le texte (un tableau édité peut comprendre 70 % de caractères blancs contigus - il est donc compressible : cf. Pom's 10 sur le compactage des écrans HGR) ?

Il est d'autre part possible d'utiliser des fichiers textes au standard DOS 3.3 mais. dans certain cas. j'ai rencontré des bugs étranges. le programme sauvant inlassablement un fichier jusqu'à ce que la disquette soit pleine. ou se "plantant" en chargeant un fichier Apple Writer... Prudence donc !

En conclusion. Magic Window II est un excellent logiciel pour la première frappe d'un document. grâce à ses excellentes qualités ergonomiques et l'utilisation très pratique du curseur qui permettent une saisie très rapide et sans migraine. mais il n'offre aucun aspect révolutionnaire dans son appproche et souffre de quelques lacunes qui rendront difficile une réécriture partielle du texte. Il reste néanmoins un produit amical. d'un rapport qualité/prix très compétitif.

NDLR - Magic Window II a également été testé par Guy Lapautre. qui complète l'analyse d'Alexandre Avrane par ces remarques :

1 - Je trouve qu'une version modernisée de ce logiciel. qui prétend être adaptée à l'Apple //e. et ce sans utiliser ses touches particulières (flèches verticales. pommes. ...). est une aberration.

2 - Les codes de déplacement utilisés (CTRL-Q ou W ou Z ou X). parfaitement ergonomiques en Qwerty. sont ridicules en Azerty (la traduction du manuel le reconnaît d'ailleurs). Je suis d'autant moins gêné pour le dire que je travaille toujours en Qwerty.

3 - Je m'oppose à l'idée d'utiliser un logiciel (Magic Window par exemple) pour une première frappe et un autre (Apple Writer par exemple) pour les corrections. Ces deux logiciels étant fondés sur des principes totalement différents. il faut de longues heures pour se débarasser des réflexes acquis par une bonne habitude d'un logiciel afin de passer à un autre.

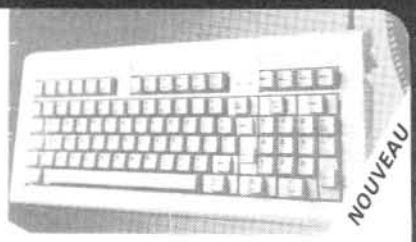
4 - Je trouve que Magic Window manque un peu de fonctionnalités aujourd'hui courantes. surtout dans le domaine des déplacements et remplacements de texte. En revanche. il a gardé une gestion très commode par menus bien conçus.

En résumé. je préférais nettement Magic Window I à Apple Writer I. Maintenant. je vote sans hésiter en faveur d'Apple Writer II. face à Magic Window II. ■

MICRO-PERIPH

ouvre les portes de votre apple II

Ouvert du mardi au samedi : 10 h - 12 h et 14 h - 18 h
62, rue Ducouédic - 75014 PARIS (Tél. : 321.53.16)



MULTITECH

clavier pour Apple

(1 515 F T.T.C.)

- 90 touches sur un clavier ergonomique et esthétique
- 12 touches de fonction progammables par l'utilisateur
- 10 touches de fonctions définissables par l'utilisateur
- 52 touches pour les commandes en Basic ou DOS
- cordon de 1,60 m
- LED pour "caps lock" et "num lock"
- parfaitement adapté pour l'Apple

Ce produit est d'une excellente qualité et durabilité. Un microprocesseur-décodeur sur le clavier confère une énorme flexibilité au niveau de la paramétrisation des touches, ce qui est très utile à l'installation d'application de traitement de texte. L'installation se fait en cinq minutes, sans soudure ni autre pièce.

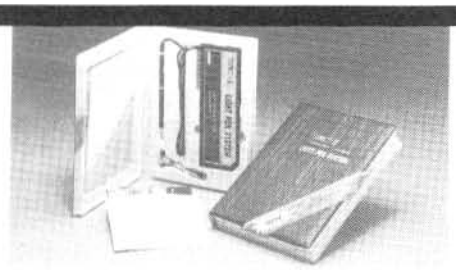
MODEM "BUZZ BOX" (1 100 F T.T.C.)

- fonctionnement aux normes françaises (CCITT)
- opération aux 300 bauds
- fonctionnement sur pile de 9 V ou avec adaptateur
- bi-directionnel avec modes "appel" et "réponse"
- full et half duplex
- livré avec notice détaillée

(Cet appareil n'a pas encore obtenu son homologation par les P.T.T.)

PROMOTION IMPRIMANTES

• Epson FX 80 F/T avec interface pour Apple	5 950 F T.T.C.
• Epson RX 80 avec interface pour Apple	4 100 F T.T.C.
• Epson RX 80 F/T avec interface pour Apple	4 450 F T.T.C.
• Epson MX 100 F/T avec interface pour Apple	6 300 F T.T.C.
• Epson FX 100F /T avec interface pour Apple	6 950 F T.T.C.
• Mannesman Spirit 80 avec interface Apple	4 500 F T.T.C.



NOUVEAU

STYLO OPTIQUE (1 770 F T.T.C.)

- fonctionne avec moniteur monochrome, couleur ou téléviseur couleur
- résolution 280 - 192 points
- 10 K disquette système qui confère 40 instructions graphiques
- fonction "ZOOM" qui élargit jusqu'à sept fois
- six couleurs de base mixables
- applications en D.A.O., animation, enseignement, etc.

LES CARTES « PILOTS »

U-TERM (1 490 F T.T.C.)

Cette carte qui vous donne 80 colonnes sur l'écran vous permet d'avoir deux jeux de caractères à la fois qui sont co-résidents dans la mémoire vive et la mémoire morte de la carte. Par exemple, les caractères français majuscules et minuscules ainsi que les caractères anglo-saxons. Elle a une très grande gamme de compatibilité : APPLESOFT, PASCAL, CP/M, APPLEWRITER II, etc.

U-Z80 (1 150 F T.T.C.) : son point fort est la vitesse : 4 MHz. Compatible avec tous les logiciels CP/M ou de Z80.

U-RAM16 (890 F T.T.C.) : c'est une carte langage entièrement compatible avec INTEGER BASIC, PASCAL, VISICALC, etc. Pourquoi payer plus cher ?

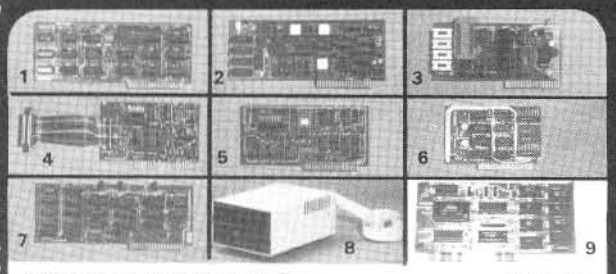
U-S232 (1 040 F T.T.C.) : une carte de série, vitesse variable entre 75 et 19200 bauds. Cette carte contient son propre logiciel de *handshake*, ce qui permet de faire fonctionner une imprimante à sa vitesse optimale. Compatible avec le logiciel de communication VISITERM.

NOUVEAU

PROMOTION DE LECTEURS DE DISQUES — 2 550 F T.T.C. (Photo 8)

- 100 % compatible avec Apple II + et //e
- Mécanique Siemens Corp. avec système vis sans fin
- Electronique fabriquée en Grande-Bretagne
- Entièrement testé et garanti 12 mois
- Opération très silencieuse et précise

(Pour les ventes par correspondance, veuillez ajouter 40 F de port.)



LES CARTES D'INTERFACE

	PRIX T.T.C.
Carte Z 80, (4 MHz) (Photo 1)	930
Carte 80 col. avec minu/majuscules français (Photo 2) car. inversés (II+) compat. Basic, Pascal, CP/M etc.	850
Programmateur d'Eproms (2758/16/32/32A/64) (Photo 3)	830
Carte RS232C 75-19200 baud compat. Visiterm, etc. (Photo 4)	720
Int/face parallèle type Centronics pour Epson, etc. (Photo 5)	475
Interface pour lecteur de disques (Photo 6)	515
Carte langage pour Apple II+ compatible Pascal, CP/M (Photo 7)	575
Carte 80 colonne pour Apple //e extensible à 64 K (Photo 9)	850
Carte 80 colonnes pour Apple //e équipée de 64 K (Photo 9)	1 400
Carte 128 K pour II+ ou //e avec pseudo disque DOS, Pascal et CP/M	2 745
Joystick avec auto-centrage et micro-adjust	275
Ventilateur pour Apple II+ ou Apple //e	320
Alimentation	650
Carte VIA 6522 avec 8 entrées/sorties et deux temporisat.	465
Carte horloge	755
Carte IEEE-488 - câble	1 650
Carte musicale	715
Carte imprim. (Eps., Cent., NEC, Appl.) a vec 64 K buffer	1 645
Carte à entre/sortie multiple A/D, D/A	2 205
Carte 6809 bi-processeur	2 205
Carte convertisseur A/O	1 085
Carte int/face para commutable (Epson, Apple, NEC, Centr.)	675
Moniteur Zenith vert	1 120
Moniteur Zenith ambre	1 230
Carte Videx Ultraterm	5 100
Accelerator II 4 MHz de Saturn - tous les programmes tournent quatre fois plus vite sans modification	4 350
Accelerator //e	4 900
SAM synthétiseur vocal	1 150
Enhancer II - buffer, auto-rep. et macros pour claviers	1 500

LES LOGICIELS PROFESSIONNELS

DBASE II	6 600
ASCII Express pro - logiciel de communication	1 550
Z-Term Pro - logiciel de communication Z 80	1 500
P-Term Pro - logiciel de communication Pascal	1 500
Softerm 2 - émulation de terminal	2 050
Magicalc	1 400
Merlin Assembleur	650

POUR OBTENIR PLUS DE RENSEIGNEMENTS, n'hésitez pas à nous appeler. D'autre part, nous avons une très grande gamme de produits pour lesquels nous pouvons donner un prix par téléphone et qui sont disponibles sous un délai de deux semaines.

BON DE COMMANDE

(Toutes nos cartes sont garanties 12 mois)

A découper et envoyer à :
MICRO-PÉRIPH, 62, rue Ducouédic - 75014 Paris - Tél. : 321.53.16
Je désire recevoir rapidement, sous pli recommandé :

QUANTITÉ	DÉSIGNATION	TOTAL T.T.C.
	Total	
	Frais de port recommandé	25,00 F
	Ci-joint un chèque / C.C.P. de F :	

NOUS ACCEPTONS DES BONS DE COMMANDE ADMINISTRATIFS

Comparaison de deux programmes Applesoft

Gérard Michel

Contrairement à ce que l'on pourrait croire, l'activité du programmeur ne s'exerce pas toujours dans le cadre d'une organisation sans faille. Peut-être faut-il voir là une manifestation de ce désordre bouillonnant qui caractérise souvent le travail des grands créateurs...

Ainsi, il vous est sans doute déjà arrivé de vous trouver avec plusieurs versions d'un même programme en Applesoft, sauvées sous des noms trop semblables pour que leurs nuances puissent vous éclairer, sans savoir discerner de façon certaine le fichier contenant les plus récentes modifications.
S'il s'agit d'un programme relativement long, la recherche des différences significatives peut représenter une perte de temps non négligeable.

La routine en langage-machine que nous vous proposons dans cet article vise précisément à limiter cette perte de temps en comparant "automatiquement" deux programmes en Applesoft chargés en mémoire, afin d'en afficher toutes les différences.
La routine est implantée à partir de l'adresse décimale 37500 et ne pourra donc être utilisée que lorsque la taille totale des deux programmes à comparer ne dépasse pas 35K environ.

Les étapes de la mise en oeuvre sont les suivantes (avec PG1 désignant la première version chargée en mémoire, et PG2 la seconde) :

- BLOAD DIF.OBJ
- LOAD PG1
- POKE 103,PEEK(175) : POKE 104,PEEK(176) : place le pointeur de l'adresse de début de programme à la fin de PG1 pour que le chargement de PG2 n'écrase pas ce dernier
- LOAD PG2
- CALL 37500

La routine peut afficher les différences suivantes :

1) Ligne numéro N listée seule en mode normal : cette ligne existe dans PG1 mais pas dans PG2.
2) Ligne numéro N listée seule en mode inverse : cette ligne existe dans PG2 mais pas dans PG1.
3) Ligne numéro N listée en mode normal, puis listée à nouveau en mode inverse : la ligne existe dans les deux programmes, mais les instructions qu'elle comporte ne sont pas strictement identiques dans les deux cas. Il suffit ainsi qu'un seul caractère soit différent pour provoquer l'affichage (nom de variable modifié, instruction rajoutée ou supprimée...).
Chaque fois qu'une ligne a été listée, qu'il s'agisse de PG1 ou de PG2, la routine attend la frappe d'un caractère au clavier avant de continuer la comparaison.

Il est également possible de lister les différences sur papier: pour ce faire, tapez PR#s (où s désigne le slot de l'imprimante) avant CALL 37500. Dans ce cas, il n'est bien sûr plus possible de répérer l'origine d'une ligne isolée par le mode d'affichage (normal ou inverse) et vous devrez utiliser la liste de l'un des programmes pour la déterminer. Lorsque deux lignes de même numéro se suivent, la première appartient toujours à PG1 et la seconde à PG2.

Si la routine ne liste rien, les deux programmes sont tout simplement rigoureusement identiques.
A l'issue de la comparaison, il peut arriver qu'un message "SYNTAX ERROR" vous soit adressé par le système. Cela provient de ce que la routine ne rétablit pas tous les pointeurs de programme à une valeur correcte avant le retour au BASIC, mais il n'y a pas à s'en inquiéter. Si vous désirez faire de suite une autre analyse, tapez NEW et reprenez le processus à partir de LOAD PG1. ■

LISA 1.5

```
 1 ;*********************************
 2 ;*                               *
 3 ;*     COMPARAISON DE DEUX       *
 4 ;*     PROGRAMMES APPLESOFT      *
 5 ;*       CODE = DIF.OBJ          *
 6 ;*                               *
 7 ;*********************************
 8        ORG $927C
 9        OBJ $800
10 TXT    EPZ $67     ;POINTEUR POUR PROGRAMME 2
11 HOME   EQU $FC58
12 GET    EQU $FD0C
13 TX0    EPZ $6      ;POINTEUR POUR PROGRAMME 1
14 INV    EPZ $32     ;DRAPEAU POUR INVERSE/NORMAL
15 IN     EQU $200    ;BUFFER D'ENTREE/CLAVIER
16 AD     EPZ $8      ;SAUV. 2 OCTETS SUIVANT LIGNE LISTEE
17 TXP    EPZ $18     ;SAUVEGARDE DE TXT
18 TX1    EPZ $1A     ;ADRESSE LIGNE SUIVANTE DANS PG1
19 TX2    EPZ $1C     ;ADRESSE LIGNE SUIVANTE DANS PG2
20 TXE    EPZ $1E     ;POINTEUR "FIN DE PROGRAMME" POUR LIST
21 CD     EPZ $EB     ;TYPE DE LIST A EFFECTUER
22        JSR HOME
23        LDA #1      ;MET TX0 AU DEBUT DE PG1
24        STA TX0
25        LDA #8
26        STA TX0+1
27        JSR S0      ;MISE A JOUR DE TX1
28        JSR S1      ;MISE A JOUR DE TX2
29 DEB    LDY #3
30        LDA (TX0),Y ;OCTET HAUT NO LIGNE DANS PG1
31        CMP (TXT),Y ; = OCTET HAUT NO LIGNE DANS PG2 ?
32        BNE C0
33        DEY
34        LDA (TX0),Y;IDEM POUR OCTET BAS NO LIGNE
35        CMP (TXT),Y
36        BEQ C3
37        BCC C02     ;NO INFERIEUR DANS PG1
38 C01    LDA #1      ;NO INFERIEUR DANS PG2
39        STA CD
40 C1     LDA #$3F    ;MODE "INVERSE"
```

```
41        STA INV
42        LDA TX2      ;POINTEUR FIN DE LIST SUR PG2
43        STA TXE
44        LDA TX2+1
45        STA TXE+1
46        JMP LIST
47 C0     BCS C01      ;NO INFERIEUR DANS PG2
48 C02    LDA #2
49        STA CD
50 C2     LDA TXT      ;SAUVEGARDE DE TXT
51        STA TXP
52        LDA TXT+1
53        STA TXP+1
54        LDA TX0      ;DEBUT DU "PROGRAMME" A LISTER
55        STA TXT      ;EST EN TX0
56        LDA TX0+1
57        STA TXT+1
58        LDA TX1      ;POINTEUR FIN DE LIST SUR PG1
59        STA TXE
60        LDA TX1+1
61        STA TXE+1
62        JMP LIST
63 C3     INY
64 C4     INY
65        LDA (TX0),Y ;COMPARAISON DES DEUX LIGNES
66        CMP (TXT),Y ;AYANT MEME NUMERO
67        BNE CB      ;SI 1 OCTET EST DIFFERENT =) LIST
68        CMP #0      ;FIN LIGNE DANS PG1 ?
69 ;(SI LA LIGNE EST FINIE AVANT DANS PG2, ON A UN 0 QUI NE SE)
70 ;(RETROUVE PAS DANS PG1 ET ON SORT POUR UN LIST DES DEUX)
71 ;(LIGNES A L'INSTRUCTION 67)
72 ;(SI LA LIGNE N'EST PAS FINIE DANS PG2, ON N'ARRIVE MEME PAS)
73 ;(A L'INSTRUCTION 68 CAR ON A DANS PG1 UN 0 QUI NE SE TROUVE)
74 ;(PAS DANS PG2 ET ON SORT A L'INSTRUCTION 67)
75 ;(LE TEST SUR PG1 SUFFIT DONC POUR VERIFIER QUE LES DEUX)
76 ;(LIGNES SONT BIEN IDENTIQUES DU PREMIER AU DERNIER OCTET)
77        BNE C4
78 C9     LDA TX1      ;PASSE A LIGNE SUIVANTE DANS PG1 ET PG2
79        STA TX0
80        LDA TX1+1
81        STA TX0+1
82        LDA TX2
83        STA TXT
84        LDA TX2+1
85        STA TXT+1
86        JSR S0
87        JSR S1
88 RD     LDA TX1+1    ;PG1 FINI ? (00 EN TX1 ET TX1+1)
89        BNE C6
90        LDA TX1
91        BNE C6
92        CMP TX2+1    ;PG2 FINI AUSSI ?
93        BNE C5
94        CMP TX2
95        BNE C5
96        LDA #8       ;REMISE DES POINTEURS EN POSITION "NEW"
97        STA TXT+1    ;ET FIN DE ROUTINE
98        STA $BD
99        LDA #1
100       STA TXT
101       LDA #4
102       STA $AF
103       RTS
104 C5    JMP C01      ;LIST DE LA FIN DE PG2
105 C6    LDA TX2+1    ;PG1 PAS FINI - PG2 L'EST-IL ?
106       BNE C7
107       LDA TX2
108       BNE C7
109       LDA #4       ;LIST DE LA FIN DE PG1
110       STA CD
111       JMP C2
112 C7    JMP DEB      ;COMPARAISON DES LIGNES SUIVANTES
113 C8    LDA #5       ;ANNONCE LIST DE 2 LIGNES DIFFERENTES
114       STA CD
115       JMP C2
116 LIST  LDX #0
117 L0    LDA INLIST,X ;MET DANS LE BUFFER "IN" LES
118       STA IN,X     ;CODES-ECRAN DE L'INSTRUCTION
119       INX          ; LIST:CALL 37716
120       CPX #$E
121       BNE L0
122       LDY #0
123       LDA (TXE),Y  ;SAUVE 2 OCTETS SUIVANT LA LIGNE
124       STA AD       ;A LISTER (DEBUT LIGNE SUIVANTE)
125       INY
126       LDA (TXE),Y
127       STA AD+1
128       DEY
129       TYA
130       STA (TXE),Y  ;MET 00 APRES LA LIGNE POUR
131       INY          ;SIMULER UNE FIN DE PROGRAMME
132       STA (TXE),Y
133       JSR $D539    ;ROUTINE D'INTERPRETATION DU BUFFER
134       JMP $D444    ;SAUT DANS LE BASIC POUR EXECUTION
135       LDY #0       ;ON EST A L'ADRESSE DECIMALE 37716
136       LDA AD       ;RESTAURE DEBUT LIGNE SUIVANTE
137       STA (TXE),Y
138       INY
139       LDA AD+1
140       STA (TXE),Y
141       JSR GET      ;DEMANDE FRAPPE AU CLAVIER
142 RL    LDA #$FF     ;RETABLIT MODE "NORMAL"
143       STA INV
144       LDA CD
145       CMP #1
146       BNE RL0
147       LDA TX2      ;PASSE A LIGNE SUIVANTE DANS PG2
148       STA TXT
149       LDA TX2+1
150       STA TXT+1
151       JSR S1
152       JMP R0
153 RL0   CMP #2
154       BNE RL1
155       LDA TXP      ;RETABLIT TXT POUR PG2
156       STA TXT
157       LDA TXP+1
158       STA TXT+1
```

```
159 RL2    LDA TX1    ;PASSE A LIGNE SUIVANTE DANS PG1
160        STA TX0
161        LDA TX1+1
162        STA TX0+1
163        JSR S0
164        JMP R0
165 RL1    CMP #4
166        BEQ RL2    ;ON EST DANS LE LIST DE LA FIN DE PG1
167        CMP #5
168        BNE RL3
169        LDA #3     ;LA LIGNE DE PG1 EST LISTEE
170        STA CD     ;ET ON VA MAINTENANT LISTER
171        LDA TXP    ;CELLE DE PG2
172        STA TXT
173        LDA TXP+1
174        STA TXT+1
175        JMP C1
176 RL3    JMP C9     ;2E LIGNE LISTEE - ON PASSE A LA SUITE
177 S0     LDY #0     ;S-P DE MISE A JOUR DE TX1
178        LDA (TX0),Y
179        STA TX1
180        INY
181        LDA (TX0),Y
182        STA TX1+1
183        RTS
184 S1     LDY #0     ;S-P DE MISE A JOUR DE TX2
185        LDA (TXT),Y
186        STA TX2
187        INY
188        LDA (TXT),Y
189        STA TX2+1
190        RTS
191 INLIST HEX CCC9D3D4BAC3C1CCCC838787B186
192        DCM "INT"
193        END
```

RECAPITULATION

```
927C- 20 58 FC A9
9280- 01 85 06 A9 08 85 07 20
9288- AE 93 20 BA 93 A0 03 B1
9290- 06 D1 67 D0 1C 88 B1 06
9298- D1 67 F0 36 90 15 A9 01
92A0- 85 E8 A9 3F 85 32 A5 1C
92A8- 85 1E A5 1D 85 1F 4C 2F
92B0- 93 B0 E8 A9 02 85 E8 A5
92B8- 67 85 18 A5 68 85 19 A5
92C0- 06 85 67 A5 07 85 68 A5
92C8- 1A 85 1E A5 18 85 1F 4C
92D0- 2F 93 C8 C8 B1 06 D1 67
92D8- D0 4E C9 00 D0 F5 A5 1A
92E0- 85 06 A5 1B 85 07 A5 1C
92E8- B5 67 A5 1D 85 68 20 AE
92F0- 93 20 BA 93 A5 1B 00 1E
92F8- A5 1A D0 1A C5 1D D0 13
9300- C5 1C D0 0F A9 08 85 68
9308- 85 B0 A9 01 85 67 A9 04
9310- 85 AF 60 4C 9E 92 A5 10
9318- D0 08 A5 1C D0 07 A9 04
9320- 85 E8 4C B7 92 4C BD 92
9328- A9 05 85 E8 4C B7 92 A2
9330- 00 BD C6 93 9D 00 02 E8
9338- E0 0E D0 F5 A0 00 B1 1E
9340- 85 08 C8 B1 1E 85 09 88
9348- 98 91 1E C8 91 1E 20 39
9350- D5 4C 44 D4 A0 00 A5 08
9358- 91 1E C8 A5 09 91 1E 20
9360- 0C FD A9 FF 85 32 A5 E8
9368- C9 01 D0 0E A5 1C 85 67
9370- A5 1D 85 68 20 BA 93 4C
9378- F4 92 C9 02 00 16 A5 18
9380- 85 67 A5 19 85 68 A5 1A
9388- 85 06 A5 1B 85 07 20 AE
9390- 93 4C F4 92 C9 04 F0 EE
9398- C9 05 D0 0F A9 03 85 E8
93A0- A5 18 85 67 A5 19 85 68
93A8- 4C A2 92 4C DE 92 A0 00
93B0- B1 06 85 1A C8 B1 06 85
93B8- 18 60 A0 00 B1 67 85 1C
93C0- C8 B1 67 85 1D 60 CC C9
93C8- D3 D4 BA C3 C1 CC CC 83
9300- 87 87 B1 86
```

Accélération de programmes en Pascal

Pour accélérer vos programmes Pascal, vous pouvez utiliser des procédures intrinsèques relativement méconnues telles que FILLCHAR et SIZEOF.

La procédure FILLCHAR permet de remplir la variable DESTINATION sur une longueur LONGUEUR avec le caractère CARACTERE; elle est définie par :

FILLCHAR (DESTINATION, LONGUEUR, CARACTERE);

DESTINATION est de type quelconque (sauf fichier); ce peut être simplement un tableau, un enregistrement ou un champ dans un enregistrement; s'il s'agit d'un tableau ou d'une chaîne de caractères, DESTINATION peut être indicée. FILLCHAR (tableau[51],50,'Z') remplit le tableau de Z de l'indice 51 à 100.

Si l'on ne connaît pas la taille exacte de la variable DESTINATION et que l'on désire la remplir complètement, on utilisera avec avantage la fonction SIZEOF qui retourne la longueur d'une variable ou d'un type (sauf fichier). Ainsi, FILLCHAR(ENREGISTREMENT, SIZEOF(ENREGISTREMENT), CHR(0)) remplit la variable ENREGISTREMENT de zéros. Le remplissage d'un enregistrement par CHR(0) correspond à '' pour les STRING, à 0 pour les INTEGER et à FALSE pour les BOOLEAN, d'où l'utilité de FILLCHAR pour obtenir rapidement et sans effort un enregistrement vierge.

Ce procédé est très rapide pour remplir de grands tableaux. Vous pourrez faire un essai en remplaçant, dans le programme saisie multipage du Pom's 9, dans la procédure INIT:

FOR I := 1 TO MAXLIGTB DO
FOR J := 1 TO MAXCOLTB DO
TB[I,J] := 0

par : FILLCHAR(TB,SIZEOF(TB),CHR(0));

Vous vous apercevrez que vous n'avez pratiquement plus le temps de lire "Patience, RAZ du tableau"! Attention cependant : le système n'exerce aucun contrôle; vérifiez bien les limites de remplissage ...

pour mieux choisir votre ordinateur et pour mieux l'utiliser :

lisez

L'ORDINATEUR INDIVIDUEL

Vous y trouverez :

• l'actualité et les tendances de l'informatique individuelle • les bancs d'essais des principaux matériels • des panoramas et des tests comparatifs • le point des grandes manifestations internationales • des articles d'initiation • des synthèses • des programmes • des interviews "exemplaires" • des conseils • des idées • des astuces

L'ORDINATEUR INDIVIDUEL, chez votre marchand de journaux

Bibliographie

Alexandre Duback

Recueil No. 2 de Pom's - Editions MEV et Editions du PSI - 215 pages - 120 FF.

Voici enfin le second recueil de Pom's, couvrant les numéros 5 à 8. Ces numéros ont déjà été retirés, mais le 5 et le 6 se trouvent à nouveau en rupture de stock.

Nous souhaitons à ce second recueil autant de succès qu'au premier. Avec les disquettes d'accompagnement, vous aurez plein de programmes, dont les "leaders" : Haifa, le HELLO complet, la programmation facilitée, le moniteur étendu, la cryptographie, de multiples utilitaires et programmes graphiques, ... Autant de programmes qui seraient normalement chacun vendus, à eux seuls, pour le prix du recueil avec ses disquettes. Bonne lecture !

Le recueil No. 2 sera disponible à la mi-avril.

Techniques de programmation des jeux, de David Lévy. Editions du PSI. 245 pages, 102 FF.

Le livre donne une bonne idée des principes de programmation d'un grand nombre de jeux de damier et de cartes. Il est dommage qu'il ne présente pas le moindre exemple de programme et seulement quelques rares organigrammes partiels. Néanmoins, un lecteur assez fort en informatique pourra en tirer profit, surtout s'il a déjà un peu travaillé sur le sujet.

Ce qui est par contre regrettable, c'est l'incroyable négligence dans la présentation de l'ouvrage : pas de table des matières, fautes d'impression nombreuses et parfois gênantes, bibliographies omises bien qu'annoncées dans le texte. La question se pose : y a-t-il un correcteur dans la maison ?

Multiplan sur Apple, par Hervé Thiriez, Editions du PSI.

Dois-je ne pas parler de ce livre puisque l'auteur participe à cette revue ? J'ai aussi un devoir d'information à votre égard: j'en parlerai donc, ayant eu la chance de pouvoir analyser le manuscrit.

Il s'agit là d'un ouvrage ayant deux buts : une introduction progressive et cependant approfondie à Multiplan, et la présentation de tableaux types et d'astuces d'utilisation nombreuses qu'ignorent parfois des utilisateurs réguliers du programme. Ces deux buts sont manifestement atteints. Nous conseillons cet ouvrage à tous les utilisateurs de Multiplan : même les habitués y apprendront des trucs.

Comme dans le cas de "Visicalc sur Apple", une disquette d'accompagnement est proposée en option. Cette disquette se vendra même toute seule, car elle contient un programme permettant d'obtenir une liste détaillée descriptive de tout tableau, comportant tous les renseignements y compris les formats, les zones "nommées", les indications de recopie, ...

Guide de l'Apple, tome 3, par Benoît de Merly - Edimicro - 180 pages, 79 FF.

On retrouve le même objectif que dans "20 progiciels-outils..." présenté dans le Pom's 10. Nous avons ici droit à l'analyse de cinq programmes de traitement de texte (surtout Applewriter et Wordstar), deux tableurs, sept gestionnaires de fichiers (trois en détail), deux logiciels de graphiques, et 18 pages sur la télématique.

Comme nous l'avons dit dans le numéro 10 de Pom's, ce type d'analyse de logiciel nous semble être ou trop ou trop peu, d'autant que, dans ce cas-ci, plusieurs logiciels sont déjà fort connus : Applewriter, PFS, CX, Visicalc, Multiplan, ...

Disk, files and printers for the Apple II, de Brian D. et George H. Blackwood. Prentice Hall International, 66 Wood Lane End, Hemel Hempstead, Herts HP2 4RG, Royaume Uni - 260 pages, $20.75.

Il s'agit là d'un ouvrage clair et facilement lisible sur le DOS, avec un chapitre bien fait sur l'utilisation de l'imprimante où l'on explique comment éviter le TAB au-delà de la quarantième colonne. Des programmes de démonstration accompagnés d'organigrammes complètent la présentation. Pour les débutants anglophones.

Programming the Apple in Basic, de Paul Tebbe, Prentice Hall, 182 pages, $22.05.

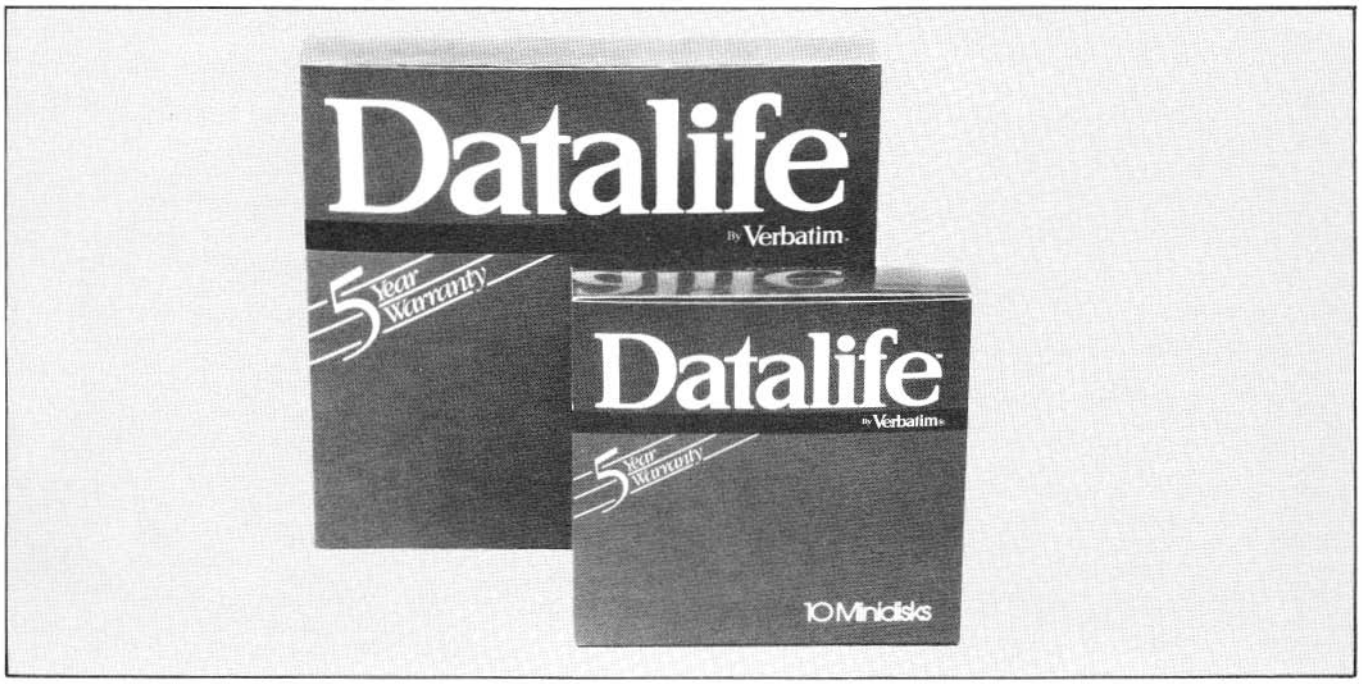
Un bon livre d'initiation au Basic sur Apple, y compris à l'utilisation des fichiers. Avantages : clarté et pédagogie. Bien entendu, il faut pouvoir lire l'anglais. Un regret : le livre ne prévient pas des limites ou bugs de l'Apple : voir à ce sujet les remarques sur l'APPEND et le RENUMBER dans le recueil No. 1 de Pom's. ■

L'augmentation du rythme de parution requiert un renforcement de notre équipe. Pom's recherche un collaborateur avec un parfaite maîtrise de l'Apple II et *//e*, capable de faire de l'assembleur les yeux fermés, connaissant les adresses de l'Apple aussi bien que celle de sa petite amie, décontracté et sympathique. Il faudrait en outre qu'il habite Paris ou le Sud-Ouest de la région parisienne (nous sommes à Versailles). S'il est par ailleurs capable d'écrire clairement et sans "fôtes d'ortografe", c'est encore mieux, mais pas indispensable. Nous sommes ouverts à toute formule entre un mi-temps et un temps plein. Avantages : travail passionnant avec des gens sympathiques (mais oui ...), dans des conditions totalement conviviales : possibilité de travailler en bonne partie chez soi, même la nuit et les week-ends. Nous cherchons aussi un responsable comptable et administratif sérieux, ordonné au point de la méticulosité : il n'est en effet pas question de perdre vos précieux bulletins d'abonnement.

Datalife

BY Verbatim

DISQUETTES ET MINI DISQUETTES TOUTES CONFIGURATIONS



- Certification unitaire 100 % sans erreur.
- Durée de vie : 30 millions de révolutions (standard de l'Industrie 3,5 millions de révolutions).
- Anneau de renforcement en standard sur le 5 1/4 ".
- 5 1/4 " en 48 et 96 TPI, simple et double face.

BFI ELECTRONIQUE - 9 RUE YVART -
75015 PARIS.
Tél. 533-01-37.

Micro-informations

Guy Lapautre
(assisté de J.-F. Duvivier)

Promenade au Jardin des Hespérides

Le cru 83 de la Pomme d'Or est composé de quatre lauréats, récompensés par des Lisa (juste avant la baisse des prix...) :

- Michel Marquis (un auteur de Pom's), avec le Kit PIA, relie son Apple à tout son appartement (comment fait-il pour se raser sans griller ses ROMs ?).
- Roland Moreno, François Grieu et Frédéric Lévy communiquent avec la carte Apple-Tell (voir plus loin).
- Claude Collin, de Contrôle X, propose CX Système (analysé dans le Pom's 10, après l'étude de CX Multigestion dans le Pom's 5).
- Jean-Luc Besnard et Patrick Choisy, avec la comptabilité Héraklès, "s'interfacent" avec Visicalc, Applewriter, Business Graphics, Omnis,...

Le "Goncourt des Logiciels" permettra-t-il une meilleure protection juridique des oeuvres des auteurs de logiciels ?

Vous avez dit compatible ?

I.E.F. propose un Apple //e portable baptisé STADU P : Système de Traitement et d'Acquisition de Données Universel Portable. On y trouve, dans un coffret robuste, un Apple //e avec un lecteur de disquettes 140 K intégré, un écran 7 pouces et un clavier ASCII abattant, le tout pour 19000 F HT. En option, on peut disposer dans ce même coffret d'une imprimante/table traçante de 115 mm (5000 F HT interface comprise), avec 16 tailles de caractères, 1 à 80 caractères par ligne, 4 sens d'écriture, 4 couleurs, 16 types de pointillés, tracé des axes X-Y, déplacements absolus ou relatifs. Il y a aussi de la place pour deux lecteurs 140 K en demi-hauteur, ou un lecteur 5 pouces 1 MO ou un disque dur 5 MO.

Toutes les cartes Apple peuvent être montées dans les connecteurs du système.

Au rayon des O.S.

Vous avez bien sûr reconnu les Operating Systems et non les travailleurs à la chaîne (fût-ce une chaîne de traitement).

Prodos va donner un coup de fouet à votre Apple II. Ce nouvel O.S. devrait être plus rapide, laisser plus de place en mémoire en se logeant dans la carte langage de l'Apple II+ ou dans les 16 K supplémentaires de l'Apple //e, et surtout ne plus limiter les volumes aux sempiternels 140 K du DOS 3.3.

Il sera facile d'utiliser un disque dur ou des disquettes de plus grande capacité. La compatibilité avec l'Apple III sera grandement améliorée. D'autres améliorations de performances sont attendues grâce à Prodos. Je cite Apple :

"... des applications avancées comme les réseaux, les graphiques professionnels, et la gestion de bases de données évoluées".

Chez le disquaire

Autour d'Apple on parle précisément de la prochaine disponibilité du disque dur Profile sur Apple II, grâce à l'interface Apple II - Profile (ce serait pour avril). Il est prévu que ce disque soit partageable entre Prodos et Pascal (mais pas CP/M...). Il est aussi question de disquettes plus musclées (capacité double des 5 pouces 1/4 actuelles).

Datalog lance le Symbfile, disque dur 5 pouces 1/4, technologie Winchester, pour Apple II et III. Il existe en 5.25, 10.5, 21 et 42 MO.

La guerre du floppy aura-t-elle lieu ? 5 pouces 1/4 traditionnels ? 5 pouces 1/4 demi-hauteur ? 3 pouces ? 3 pouces 1/2 ? Les mini-formats continuent leur développement. Par exemple, Hitachi annonce le 3 pouces MD3, 500 KO, à moins de 3000 F, avec contrôleur (standard Apple). Mais les disquettes coûtent relativement cher : 65 F pièce. Après expérimentation, un banc d'essai pourra figurer dans un prochain numéro de Pom's (Acer Micro).

Apple, de son côté, a adopté le 3 pouces 1/2 au standard Sony, qui équipe Mac, et maintenant Lisa. Indiscrétion... il devrait être proposé sur la "ligne 6502" vers la fin de l'année.

BIMP annonce une disquette 5 pouces 1/4 de 1 MO formaté (1,6 MO non formaté), compatible MEM.DOS, DOS 3.3, CP/M et Pascal. Pour 13000 F TTC.

Impression

Les cartes "buffer" d'imprimante deviennent monnaie courante. Pentasonic propose 2 modèles : 16 K et 128 K, aux prix respectifs de 2310 F et 3970 F. De son côté, MID propose 64 K pour 2200 F, tandis qu'il en coûte 1750 F pour avoir 16 K chez Acer Micro.

Chez Apple, la nouvelle Imagewriter remplace, sur la gamme 68000, l'actuelle Dot Matrix. Elle est également proposée sur la gamme 6502.

Une naissance aussi chez Seikosha : dans la famille GP, la toute petite GP 50 A, graphique, pour 1250 F.

Nous n'avons pas encore pu essayer la nouvelle Canon couleur, ni la TRS 80 CGP 220 sept couleurs, toutes deux à jet d'encre, et dont on dit merveilles.

Signalons en vrac la FX100, haut de gamme Epson, la Gemini 10 X et la Delta 10 de Hengstler. Facit propose une haut de gamme matricielle de qualité courrier, la 4570 (2 versions à 40800 et 33700 F) et le "petit" modèle 4512, avec de bonnes fonctionnalités graphiques (pour 10080 F TTC).

Communiquons

On peut, en particulier, communiquer avec Apple-Tell (voir ci-dessus la Pomme d'Or) qui offre pour 5900 F. HT :

- un modem avec cinq vitesses de communication (half duplex en 600 et 1200 bauds et full-duplex en 300 bauds - canal de retour 5 et 75 bauds (norme Minitel) en 1200 bauds - six modes de fonctionnement CCITT (normes françaises et européennes), cinq modes de fonctionnement "Bell" (normes US), une grande variété de formats et des boucles de jonction analogique et digitale (types 2 et 3) permettant le test de la carte, de la connexion au réseau, à la ligne téléphonique et à un modem distant.
- une interface réseau avec circuit de numérotation décimale programmable et détection de sonnerie selon les formats US et européens.
- un générateur Vidéotext : buffer de 4 Ko, écran de 24*40, gestion complète du protocole Télétel.

- une interface vidéo avec sortie sur prise Péritel, haute qualité couleur, N/B Vidéotext et N/B Apple standard avec commutation automatique.
- le "firmware" : interface Apple. valeurs par défaut en mémoire EPROM permettant de mettre jusqu'à 256 critères différents à la mise sous tension.
- une disquette d'accompagnement et une documentation claire rendant ces possibilités accessibles au non-spécialiste et contenant une émulation du programme terminal Minitel.

Apple Bus se précise... pour fin 84. 16 matériels Apple (tous sont acceptés), partage de ressources (imprimante, disque de grande capacité,...), 250 Kbauds sur 300 mètres. Les moyens de connexion sont déja intégrés dans Mac et Lisa.

Dialogue

Pour "parler" à votre Apple II, le clavier d'origine va devenir un jour aussi démodé que l'est le DOS 3.3 !

Si vous tenez à la solution clavier, un clavier avec bloc numérique séparé vous coûtera environ 1000 F.

Touchez votre écran avec un crayon lumineux. Cette baguette magique, avec son logiciel, se trouve aux environs de 5000 F.

Plus rustiques, mais aussi beaucoup moins chers, les "joysticks" - disons manches à balai. Il en existe de nombreux modèles entre 200 et 800 F.

Vous pouvez préférer un système plus sophistiqué comme le Robo 1000 de Sideg (ensemble levier de commande et logiciel de dessin assisté par ordinateur), mais pour 7940 F TTC.

Une mini tablette graphique vous permet aussi de communiquer aisément avec votre Apple favori. N'en n'attendez pas des miracles de précision, mais elle ne coûte que 1000 F environ. Un peu plus ambitieuse, une série Koala Pad, à partir de 1300 F chez SPID Micro. G3I propose un système original : Graphiscop, fonctionnant avec un crayon à bille... et quand même un petit appareil (990 F TTC).

Enfin, par une astucieuse, mais coûteuse adaptation, on commence à pouvoir transformer son écran banal en écran tactile.

Tout ceci, bien entendu, en attendant la désormais traditionnelle souris ... annoncée par Apple. Je cite :

"...inaugurée avec Lisa, la généralisation de la technologie graphique souris à l'ensemble des systèmes Apple...". Indiscrétion : ce sérait pour avril, et il en coûterait moins de 1000 F, interface et logiciel inclus.

Si vous préférez des échanges par capteurs et actionneurs, Créatic vous propose un système d'interfaçage (CM 1000) avec nombreuses cartes d'interface analogiques et numériques. 1600 F TTC le coffret, 600 à 900 F par carte, 350 F pour un capteur.

Et les logiciels ?

La tendance forte est la francisation des logiciels généraux Anglo-Saxons. Les premières réalisations étaient essentiellement des traductions de la documentation. De plus en plus aujourd'hui, c'est le logiciel lui-même qui est francisé.

Un bon exemple est donné par Apple Writer II. Il montre un effort réel : tous les messages sont rédigés en Français: il montre aussi une limite : les mnémoniques de commande n'ont pas changé, et 'L' veut toujours dire "charger"...

Un nouveau-né dans la famille PFS : PFS Write, traitement de texte pouvant exploiter les fichiers, tableaux et graphiques de ses aînés. Il nous a été présenté à la boutique Galilée Informatique.

Toujours en traitement de texte, Micropro diffuse Wordstar 3.3 (déja présent au SICOB), avec fusion de fichier et utilisation rationnelle de touches de fonction. Il existe pour Apple II et III et pourra prochainement être associé à Starindex (générateur de table des matières) et Starbust (générateur de menus).

En version 2, le traitement de texte Epistole de Micro Assistance est doté de possibilités de calcul. Il fonctionne sur Apple II et Apple //e. De son côté, Gutenberg de Micromation Limited (analyse détaillée dans ce numéro) permet d'utiliser, de composer, des graphismes originaux tels que symboles mathématiques, chimiques, électroniques, etc.

En gestion de fichier, Logifiche 2 est développé par la CFNI. Nombreuses possibilités de calcul, de traitements automatiques, d'éditions et de recherche multicritères (1420 F, pour Apple II et //e).

Côté graphique, Charts Unlimited permet de créer des schémas, graphiques, organigrammes, plans... Simple à utiliser, facile à apprendre, ce logiciel n'est à notre connaissance pas encore commercialisé en France. Produit par Graphware, distribué en Europe par Transatlantic Software. Prix US : 195 $. Une analyse détaillée sera proposée dans le prochain numéro de Pom's.

B.I.P. distribue de son côté plusieurs programmes graphiques : Koalapad (moins de 1500 F TTC pour logiciel + tablette graphique) est une "ardoise magique" pour dessiner, décalquer, colorier, de façon simple mais cependant riche et variée. TGS (compatible avec le précédent, 1695 F) est un bon logiciel pour la création de dessins animés.

A quand un "Lotus 1 2 3" ou un "Visi On" sur Apple II? Sans doute aussi pour avril, sous forme d'un logiciel intégré (gestion de fichier +tableur + traitement de texte) commandé par souris, exigeant 64 K supplémentaires. Ce logiciel sera proposé par Apple sur la ligne 6502.

Côté Apple ///, SANE (Standard Apple Numeric Environment) permet des traitements numériques précis. Apple /// Pascal Toolkit est un nouvel outil d'aide au développement de programmes.

Une initiative intéressante d'Axone : l'établissement d'un "catalogue" de logiciels professionnels dont la vente est assortie d'une formation. Dans ce catalogue, pour Apple II, on relève :

- Décisionnel graphique de Adde Marketing (analyse dans le précédent numéro de Pom's).
- Incredible Jack de Business Solutions (certains font la fine bouche, mais c'est intéressant pour qui n'a pas d'ambitions démesurées).

Où vont les logiciels ?

Selon une enquête de Computer Merchandising International (janvier 84), les "best sellers" dépendent largement du pays. Par exemple, l'utilisateur Allemand d'Apple porte essentiellement son choix sur Multiplan, Wordstar et Visicalc. L'Anglais est sensible au charme de Multiplan et Wordstar (comme son collègue germain) mais aussi d'Applewriter et d'Omnis. Le Français papillonne de Multiplan à Visicalc, de Dbase II à CX Multigestion, PFS et Omnis, sans négliger Applewriter.

De source Apple généralement bien informée, l'évolution des logiciels généraux se dessine de la façon suivante :

- Pour les logiciels scientifiques et techniques, accent sur le DAO (dessin assisté par ordinateur), le traitement de texte avec caractères spéciaux (voir notre analyse de Gutenberg), et l'interfaçage avec les logiciels graphiques.
- Côté gestion, le graphique souris, le pont des logiciels intégrés, un effort considérable sur documentation et formation.
- L'éducation cherche des logiciels d'EAO commodes (Logo, Super-

pilot, EVA), a besoin de produits français originaux et s'oriente vers la variété du dialogue : vidéo-disque, tablette graphique, joystick,...

- L'utilisateur "familial" recherche quelques produits simples et bon marché : un traitement de texte à 500 F, une collection de modèles financiers à 400 F et un logiciel graphique couleur à 500 F (plus des jeux et des programmes éducatifs).

A signaler qu'Apple, en concertation avec la Datar, a choisi la ville de Metz pour installer son centre français de développement et d'adaptation de logiciels (ouverture premier semestre 1984).

Où faire son marché ?

Une nouvelle boutique lancée à grand renfort de publicité : le Multistore de Hachette, récemment ouvert à l'Opéra. Micro-ordinateurs, calculettes, librairie, vidéo et son... J'y ai vu un Apple II avec un lecteur de disquettes et un écran N. et B., ni plus ni moins bien traité que la vingtaine de modèles exposés. La programmathèque est bien fournie en logiciels de jeux, mais semble beaucoup plus discrète sur les logiciels professionnels.

Chez International Computer, des services personnalisés aux acheteurs de micros (dont Apple) : extension de garantie à 2 ans renouvelables, service de maintenance sur place (Paris et couronne), livraison à domicile des consommables.

Pentasonic propose un dépannage "immédiat" ou le prêt d'un matériel équivalent, ainsi qu'une formule d'essai de logiciel à domicile.

JCR pratique une autre formule : le "libre-service" du logiciel.

Micro Case se pose en spécialiste de la boutique d'occasions sélectionnées.

Adresses

Acer Micro - 42 rue de Chabrol - 75010 Paris - Tél 770.28.31
Adde Marketing - 27 rue des Vignes - 75016 Paris - Tél 527.90.17
Axone - Tour Neptune - 92086 Paris La Défense - Tél 773.63.64
BIMP - 20 rue Servient - 69003 Lyon - Tél (7)860.84.27
BIP - 22 rue Joseph Dijon - 75015 Paris - Tél 255.44.63
Business Solutions - 60 East Main street - Kings park - Middletown - Ohio 45042 - USA
Controle X - Tour Maine-Montparnasse - 33 avenue du Maine - 35755 Paris Cedex 15 - Tél 538.98.87
Datalog - 87 rue Lemercier - 75017 Paris - Tél 228.14.18
G3I - 5 passage Courtois - 75011 Paris
Galilée Informatique - 45 rue Galilée - 75116 Paris - Tél 723.33.30
IEF - 228 rue Lecourbe - 75015 Paris - Tél 828.06.01
International Computer - 29 rue de Clichy - 75009 Paris - Tél 285.24.55
JCR - 58 rue Notre-Dame de Lorette - 75009 Paris - Tél 282.19.80
Micro-Assistance - 66 rue Castagnary - 75015 Paris - Tél 530.05.28
Micro Case - 46 rue de La Rochefoucault - 75009 Paris - Tél 874.05.20
MID - 51 bis avenue de la République - 75011 Paris - Tél 357.83.20 - et, à partir de mars, nouvelle adresse : 96 boulevard Richard Lenoir - 75011 Paris - Tél sans changement
Pentasonic - 5 rue Maurice Bourdet - 75016 Paris - Tél 336.26.05
Sideg - 170 rue Saint Charles - 75015 Paris - Tél 557.79.12
SPID Micro - 39 rue Victor-Massé - 75009 Paris - Tél 281.20.02
Transatlantic Software - Branch Lotspeich - PO box 29561 - Cincinnati - Ohio 45229 - USA ■

Un "type-ahead" en Applesoft

Olivier Herz

Ceux d'entre vous qui ont pratiqué Visicalc ou le Pascal Apple ont sans doute remarqué que, quand ils tapent des touches pendant que la machine "réfléchit", ces touches sont prises en considération. Nous voulons prouver dans ce petit article que la même chose est possible en Applesoft.

Premier problème : stocker les touches

Il s'agit de récupérer le plus rapidement possible les touches enfoncées et de les stocker dans une table. Pour cela, il faut opérer un "aiguillage" dans l'interprétation des programmes Applesoft. Comme ce langage est généralement en ROM, le plus commode est de modifier sa seule routine en RAM, à savoir CHRGET :

```
00B1-E6 B8 CHRGET INC TXTPTR
00B3-D0 02 BNE CHRGOT
00B5-E6 B9 INC TXTPTR+1
00B7-AD YY XX CHRGOT LDA
$XXYY
00BA-C9 3A CMP #':'
00BC-B0 0A BCS FIN
00BE-C9 20 CMP #' '
00C0-F0 EF BEQ CHRGET
00C2-38 SEC
00C3-E9 30 SBC #'0'
00C5-38 SEC
00C6-E9 D0 SBC #$D0
00C8-60 FIN RTS
```

Cette routine, qui est un modèle d'astuce, incrémente TXTPTR ($B8-B9) et charge l'accumulateur A avec TXTPTR. De plus, Z est mis à 1 si A est ":" ou une fin de ligne ($3A ou $00). Et C est mis à 0 si A est un nombre ASCII ("0" à "9").

Ce que nous allons faire, c'est mettre un "aiguillage" JMP AHEAD au début de CHRGET en direction de la routine AHEAD qui teste si le clavier est enfoncé et, le cas échéant, stocke la touche dans une table (exceptions : CTRL-X annule le buffer de stockage des touches; CTRL-C fait de même, mais ne remet pas le clavier à zéro, afin de laisser interrompre éventuellement le programme). A la fin de cette routine, on incrémente TXTPTR et on saute à CHRGOT.
Quand la table de stockage est pleine, chaque fois qu'on ajoute une touche supplémentaire, la touche la plus ancienne est "oubliée" faute de place.

NB - Les spécialistes auront remarqué que le logiciel MEM/DOS lui aussi, pour permettre ses commandes LET..., a besoin d'utiliser un aiguillage dans CHRGET. Contrairement à ce que l'on pourrait croire, ce détour dans une routine très fré-

quemment utilisée comme CHRGET ralentit très peu le programme.

Second problème : récupérer les touches

Cette fois-ci, on fait un "aiguillage" dans la routine d'entrée de caractères. On récupère l'adresse de cette routine en $AA55-AA56 ($6A55-6A56 pour un 32K), où on la remplace par l'adresse de notre routine de récupération du "type ahead".
Dans cette routine, si la table de stockage est vide, on saute à la routine d'entrée de caractères. Sinon, on récupère une touche de la zone de stockage.

Attention : le DOS ne doit pas avoir été déplacé sur la carte langage.

NB : on ne peut pas utiliser $38-39 au lieu de $AA55-AA56, dans le cas d'un aiguillage, comme on le fait dans le cas de la reprogrammation de la sortie de caractères.

Gestion de la table de stockage

Cette table est située à la fin de la routine. Elle est de type FIFO (first in-first out : premier entré-premier sorti), à différencier d'une pile, de type LIFO (last in-first out : dernier entré-premier sorti). On utilise pour cela 2 pointeurs, qui se déplacent de façon circulaire dans la table : s'ils débordent à droite, ils reviennent à gauche. Le pointeur DEBUT, comme son nom l'indique, représente le début du stockage, et le pointeur FIN représente la position faisant suite à la fin du stockage. Par conséquent, quand ces pointeurs sont égaux, il n'y a pas de touches stockées. Enfin, on se doute que l'adressage des valeurs contenues dans la table se fait par indexation avec ces deux pointeurs. Par conséquent, la table ne peut avoir plus de 256 caractères.

Utilisation et limitations

Ce programme peut être assemblé n'importe où et n'utilise aucune mémoire en dehors de lui-même. Nous fournissons une version assemblée à l'adresse 768 ($300) avec une table de 60 caractères.

Pour le connecter, il suffit de le BRUNner ou de faire un BLOAD et CALL 768. Pour le déconnecter, il suffit de reprogrammer la routine d'entrée de caractères, ou de faire un PRINT CHR$(4)"IN#0", ou encore d'appuyer sur RESET. En fait, on ne déconnecte que la récupération du "type ahead", et non le stockage des touches qui continue, mais reste

LISA 2.5

```
 1 ;************************
 2 ;*                      *
 3 ;* TYPE AHEAD EN APPLESOFT *
 4 ;*                      *
 5 ;* (C)  D. HERZ POUR POM'S *
 6 ;*                      *
 7 ;************************
 8 ;
 9          ORG $300
10          OBJ $300
11 ;
12 ;   ADRESSES UTILISEES
13 ;
14 CHRGET   EPZ $B1          ;LECTURE D'UN OCTET DU PROGRAMME
15 CHRGOT   EPZ $B7          ;IDEM SANS INCREMENT DE TXTPTR
16 TXTPTR   EPZ $B8          ;POINTEUR DU PROGRAMME
17 KBD      EQU $C000        ;CLAVIER
18 KBDSTRB  EQU $C010        ;RAZ CLAVIER
19 DOSKSWL  EQU $AA55        ;$6A55 POUR 32K
20 ;
21 ;   AIGUILLAGE DE CHRGET
22 ;
23          LDA #$4C         ;JMP AHEAD EN $B1
24          STA CHRGET
25          LDA #AHEAD
26          STA CHRGET+1
27          LDA /AHEAD
28          STA CHRGET+2
29 ;
30 ;   AIGUILLAGE ENTREE CARACTERES
31 ;
32          LDA DOSKSWL      ;ON STOCKE L'ANCIEN KSWL
33          STA ANCIEN+1
34          LDA DOSKSWL+1
35          STA ANCIEN+2
36          LDA #RECUP       ;AIGUILLAGE DU KSWL
37          STA DOSKSWL
38          LDA /RECUP
39          STA DOSKSWL+1
40          LDA FIN
41          STA DEBUT
42          RTS
43 ;
44 ;   REMPLISSAGE DU TYPE AHEAD
45 ;
46 AHEAD    LDA KBD          ;TOUCHE ENFONCEE?
47          BPL )1
48          CMP #$83         ;CTRL-C
49          BEQ )4
50          BIT KBDSTRB
51          CMP #$98         ;CTRL-X
52          BNE )3
53 )4       LDA FIN          ;ON ANNULE LE BUFFER
54          STA DEBUT
55          JMP )1
```

Employons bien les BOOLEANS

On lit trop souvent dans les programmes Pascal des lignes telles que :
IF X=1 THEN BON := TRUE ELSE BON := FALSE
directement dérivées du BASIC.

Ce type de ligne s'écrit plus simplement en Pascal avec :
BON := X = 1;

```
56 ^3        STX STOCKEX
57           LDX FIN
58           STA TABLE,X          ;ON STOCKE LA TOUCHE
59           JSR INCRX
60           STX FIN
61           CPX DEBUT            ;BUFFER PLEIN?
62           BNE )2
63           JSR INCRX
64           STX DEBUT
65 ^2        LDX STOCKEX
66 ^1        INC TXTPTR           ;CODE OU DEBUT DE CHRGET ($81-$B6)
67           BNE )0
68           INC TXTPTR+1
69 ^0        JMP CHRGOT
70 ;
71 ;   RECUPERATION DU TYPE AHEAD
72 ;
73 RECUP     STX STOCKEX
74           LDX DEBUT
75           CPX FIN              ;BUFFER DE TYPE AHEAD VIDE?
76           BNE )0
77           LDX STOCKEX
78 ANCIEN    JMP $0000            ;ON SAUTE A L'ANCIEN KSWL
79 ^0        STA ($28),Y          ;-) NOP NOP POUR CERTAINS KSWL
80           LDA TABLE,X          ;ON RECUPERE UNE TOUCHE
81           JSR INCRX
82           STX DEBUT
83           LDX STOCKEX
84           RTS
85 ;
86 ;   INCREMENTE X CIRCULAIREMENT
87 ;
88 INCRX     INX
89           CPX #TAILLE
90           BNE )0
91           LDX #0
92 ^0        RTS
93 ;
94 ;   OCTETS DE STOCKAGE
95 ;
96 STOCKEX   HEX 00               ;POUR NE PAS PERDRE LE REGISTRE X
97 DEBUT     HEX 00               ;INDICE DU DEBUT DU STOCKAGE
98 FIN       HEX 00               ;INDICE DE LA FIN DU STOCKAGE
99 TAILLE    EPZ 60               ;TAILLE DU TABLEAU DE STOCKAGE
100 TABLE    DFS TAILLE           ;TABLEAU DE STOCKAGE A LA FIN DU PROGRAMME
101          END
```

Récapitulation

```
0300- A9 4C 85 B1 A9 29 85 B2
0308- A9 03 85 B3 AD 55 AA 80
0310- 77 03 A0 56 AA 8D 78 03
0318- A9 68 8D 55 AA A9 03 8D
0320- 56 AA AD 92 03 8D 91 03
0328- 68 AD 00 C0 10 31 C9 83
0330- F0 07 2C 10 C0 C9 98 D0
0338- 09 AD 92 03 8D 91 03 4C
0340- 5F 03 8E 90 03 AE 92 03
0348- 9D 93 03 20 8B 03 8E 92
0350- 03 EC 91 03 D0 06 20 B8
0358- 03 BE 91 03 AE 90 03 E6
0360- B8 D0 02 E6 B9 4C B7 00
0368- BE 90 03 AE 91 03 EC 92
0370- 03 D0 06 AE 90 03 4C 00
0378- 00 91 28 BD 93 03 20 B8
0380- 03 BE 91 03 AE 90 03 60
0388- E8 E0 3C D0 02 A2 00 60
```

ignoré. Pour déconnecter aussi le stockage, il suffit de rétablir CHRGET, soit en réinitalisant l'Applesoft (avec FP, CTRL-B ou 3D3G depuis le moniteur), soit en créant une routine en assembleur qui rétablit CHRGET (mais surtout pas avec des POKEs ! Pourquoi ?)

Ce "type ahead" est pratiquement compatible avec tout programme Applesoft à quelques restrictions près :

- Il ne doit bien sûr pas se trouver au même endroit qu'une routine assembleur utilisée par le programme (entrée par BLOAD, par DATA et POKE, ou bien par la routine de S.H. LAM, ou une routine "cachée" à la fin du programme).

- Si le programme Applesoft lit directement le clavier par PEEK(−16384), il est évident que le "type ahead" ne sera pas pris en compte, mais attendra le prochain GET ou INPUT, ou la fin du programme et le retour au niveau de commande de l'Applesoft, toutes opérations qui utilisent la routine d'entrée de caractères.

- Si le programme Applesoft reprogramme la sortie de caractères, il faut refaire un CALL 768 après cette reprogrammation (si 768 est l'adresse où l'on a assemblé le "type ahead"). Dans la majorité des cas, il ne devrait pas y avoir de conflit entre la routine reprogrammée et le "type ahead".

- Ce "type ahead" est totalement allergique à PLE : il ne marche pas si PLE est présent. De plus, si le "type ahead" est actif, le chargement de PLE peut conduire à une catastrophe.

- Les touches sont stockées au cours d'un CHRGET, c'est-à-dire quand l'interpréteur progresse dans le programme. Il est donc inutile d'espérer stocker des touches pendant les appels à des routines assembleur ou, ce qui revient au même, pendant un LIST, un PRINT d'une chaîne de caractères, ou encore un ordre DOS..., toutes ces routines n'utilisant pas CHRGET.

- Attention à ne pas initialiser deux fois consécutives le "type ahead" : sinon l'aiguillage de la routine d'entrée de caractères va se "mordre la queue" et il faudra appuyer sur RESET.

Remarque : j'ai découvert après avoir rédigé cet article que le Diversi-DOS, un système d'exploitation rapide du commerce, possède une routine de "type ahead" utilisant la même approche, mais qui marche aussi sous DOS et en mode moniteur. Bien entendu, ce programme est plus gros et plus complexe.

■

Courrier des lecteurs

Olivier Herz

*1 - Dans le programme Hello de la page 26 (Pom's 6). Je n'ai pas réussi à faire tourner le programme. Comment définissez-vous "*AF: FE0C" ?*

2 - Avec la fonction EXEC. INIT me donne une disquette initialisée qui ne rend plus la main à l'écran : curseur rectangle clignotant.

Andre Van Eggelpoel - 4 rue Porchefontaine - 94370 Sucy en Brie

1 - L'adresse $0CFE, qu'on met en $AF,B0 (par *AF:FE 0C) dans le programme HELLO a été trouvée en assemblant le programme : c'est l'adresse de la fin du programme assembleur. En mettant cette adresse en $AF,B0 qui est l'adresse de la fin du programme Applesoft, on peut cacher le programme assembleur dans le programme Applesoft.

2 - INIT ne fait pas de bien belles choses avec un fichier EXEC, car ce fichier va chercher les commandes à effectuer sur la disquette qui, après initialisation, ne contient que des "0".

Je n'ai aucune formation informatique. Je pourrais être un Appleomane presque heureux, mais voici que se pointe le langage machine, dont je ressens à travers la lecture de Pom's l'utilité et la nécessité. Je m'attèle à la tâche avec persévérance. J'avoue que les résultats ne sont pas toujours encourageants et regrette l'absence de livres ou de manuels assez clairs pour être compris d'emblée par les néophytes... Cherchant à initialiser ma Seikosha GP 100A en langage machine, j'ai essayé :

1 - de mettre directement en $36-37 : 00C1: l'imprimante s'initialise effectivement, mais se borne à imprimer le "" du moniteur;*

2 - de faire un petit programme, qui hélas donne le même résultat que la tentative précédente :

```
300: LDA #00
LDX #C1
STA $36
STX $37
RTS
300G
```

3 - en remplaçant le RTS ci-dessus par BRK, l'imprimante réagit toujours en imprimant "" ainsi que le contenu des registres qui apparaissent à la suite du lancement de ce microprogramme, puis plus rien.*

L'ennui, c'est que depuis mes essais je n'arrive plus à initialiser l'imprimante avec le classique "1 CTRL-P" à partir du moniteur, alors que le "PR#1" en Basic est toujours opérationnel.

R. Burger - 67, rue Chauzy - 51800 Ste. Menehould

En réponse à votre lettre, voici quelques explications sur la reprogrammation des entrées-sorties sur l'Apple.

a) Lorsque le DOS est absent :

Les mémoires CSW ($36-37) et KSW ($38-39) contiennent l'adresse des routines de sortie et d'entrée de caractères. Par défaut, à l'initialisation, ces adresses sont celles des routines du moniteur COUT1 ($FDF0) et KEYIN ($FD0C). Pour changer les

DONNEZ DU RÉPONDANT A VOTRE APPLE

IMAGOL présente
MEM/TERM

MEM/TERM est un logiciel de communication transformant un APPLE en terminal d'un APPLE II ou III (transmission par ligne directe ou réseau P et T)

Applications

— Exécution de programmes à distances.
(Pour tous les programmes sous MEM/DOS 6 502).
— Télémaintenance, diagnostic à distance.
— Correction de programmes à distance.
— Démonstration de programmes à distance.
— Edition à distance, etc...

NOUVEAU

CONFIGURATIONS REQUISES
Poste Principal :
• 1 unité de disque, une carte Super-Sérial.
• 1 modem électrique ou acoustique.
Poste Terminal :
• 1 unité de disque, une carte Super-Sérial.
• 1 modem electriclue ou acoustique avec porteuse.
• Transmission par le réseau commuté.

REVENDEURS CONSULTEZ-NOUS.

IMAGOL
L'artisan de votre réussite en micro-informatique professionnelle.

1 à 5, rue GUTENBERG,
75015 PARIS.
Tél. : (1) 577.59.39.

Réactions 662 46 47

routines d'entrées-sorties, il suffit de changer CSW ou KSW, soit par des POKEs, soit par un programme assembleur comme le vôtre.

NB - Pour déconnecter le DOS, il suffit de faire ":PR#0:IN#0" sous l'Applesoft (cf. Pom's 7 page 26 : trucs et astuces).

b) Lorsque le DOS est présent :

CSW et KSW contiennent les adresses des routines du DOS qui s'occuppent des entrées-sorties ($9EBD et $9E81 pour un DOS 48K; remplacer $9E par $5E pour un 32K).Ces routines font, après traitement par le DOS, un saut aux véritables routines d'entrée et de sortie (KEYIN et COUT1 par défaut après le boot), dont le DOS stocke les adresses en $AA53-AA54 (vrai CSW) et $AA55-AA56 (vrai KSW). Notons que $AA.. devient $6A.. pour un DOS 32K.

Il ne suffit plus, pour changer les entrées-sorties, de changer CSW et KSW, car le DOS "trafique" ces adresses. Mais, au contraire, il y a deux solutions possibles équivalentes :

- on fait suivre le changement de $36-37 ou $38-39 par un appel à INTDOS (CALL 1002 en Applesoft ou JMP $3EA en assembleur), qui est la routine du DOS qui reprend en main les entrées-sorties. Alors, le DOS rétablit "ses" adresses ($9E81 et $9EBD) en CSW et KSW et remet les valeurs que l'utilisateur y avait mises, en $AA.. Cette méthode présente l'intérêt de fonctionner en 48K ou 32K.

- on change directement les valeurs des "vrais" CSW et KSW ($AA..). Dès lors, cela n'a plus rien d'étonnant que vos essais ne marchent pas. De plus, votre remarque sur "1 CTRL-P" et "PR#1" s'explique très bien : faire "1 CTRL-P" depuis le moniteur ou ":PR#1" pour l'Applesoft (voir Pom's 7 page 26 pour l'utilité des deux-points) ne fait que changer CSW et donc ne marche que si le DOS est déconnecté (de même, "1 CTRL-K" et ":IN#1" ne changent que KSW). Au contraire, le "PR#1" et le "IN#1" du DOS, que l'on peut faire sous Applesoft aussi bien que sous moniteur, font les choses correctement comme indiqué ci-dessus.

Pourriez-vous me dire s'il existe une solution pour dépasser le nombre maximum de 105 titres au catalogue sur Apple ? Dans le cas qui m'intéresse, il s'agit de l'exploitation d'une multitude de petits fichiers en Applewriter II.

Docteur J.-C. Selosse - 1 avenue Carnot - 62120 Aire sur la Lys

Malheureusement, il n'est pas possible de mettre plus de 105 fichiers dans un catalogue, sauf à modifier la façon dont fonctionne le DOS. Ce que nous pouvons vous conseiller le cas échéant, c'est de regrouper plusieurs petits fichiers Applewriter en un seul.

Vous mettez alors ".ep0" au début de chaque fichier, pour bloquer l'impression du fichier. Pour imprimer le texte correspondant à un de ces fichiers originaux, il suffira de remplacer ".ep0" par ".ep1" au début du texte avant de lancer l'impression. Cette recette fonctionne avec Applewriter II ou //e.

J'ai bien reçu le dernier numéro de Pom's avec sa disquette. En essayant le calendrier sur Visicalc, j'ai eu la surprise de constater un décalage des jours de la semaine à partir du 3 janvier 84, alors que tout est normal pour une année non bissextile. Je pense qu'il faut remplacer, dans la formule de J7, la cellule B6 par la cellule D6, afin d'obtenir :

>J7: + N6 – L6 +@IF(@AND(I7, D6 < 3), 32, 33)

Vous m'obligerez en confirmant le bien-fondé de cette correction.

Claude Coviaux - 351 Bd. Pasteur - 59500 Douai

Vous avez parfaitement raison, ainsi que les nombreux autres lecteurs perspicaces ayant émis la même remarque : ce bug nous rassure d'ailleurs beaucoup sur Olivier Herz. Nous en étions arrivés à nous demander s'il n'était pas un extra-terrestre. Mais, puisque "errare humanum est", tout est pour le mieux dans le meilleur des mondes !

L'informatique peut donner des applications intéressantes dans le domaine de la physique. J'ai écrit un programme qui visualise de manière bien animée des ondes sur l'écran, ainsi que leurs interférences. Les ondes sont définies par l'utilisateur à l'aide de fonctions, ou choisies parmi les vingt exemples proposés. Ce même programme permet aussi de bien comprendre ce que signifie la mise à la vitesse v d'une onde stationnaire, selon les transformations de Lorentz. Ceci met en évidence le rapport qui existe entre les ondes et la relativité.
J'ai également écrit un second programme sur les transformations de Lorentz, base de la relativité. Il permet de visualiser le comportement des règles et horloges des systèmes de référence soumis à ces transformations, et de mieux comprendre la signification de ces formules.
La relativité et la mécanique des ondes ne sont pour vous que des formules compliquées ! Voici des animations et des visualisations qui vous permettront de mieux les aborder. Les programmes sont longs (16Ko chacun), le premier (Ondes et Relativité) étant pour moitié en langage machine, animation oblige.

Serge Cabala - 13 rue du Pied de Roche - 25660 Saône

Nous avons testé ces programmes qui sont maintenant bien au point, après un va-et-vient avec leur auteur. Nous ne pouvons les publier dans Pom's, compte tenu de la place qu'ils prendraient et de leur spécificité. Contre un chèque de 100 FF pour remboursement de ses frais, l'auteur vous enverra les programmes sur une disquette, avec deux articles d'accompagnement (14 pages).
Jmœ

Lecteur très intéressé de Pom's, car utilisateur de l'Apple II, je suis tombé sur un os. On m'avait posé la question suivante : comment retrouver la position du curseur en HGR, pour balayer l'écran en testant la présence d'un dessin ?

Exemple : un fichier de forme, ayant été chargé, dessine une figure, puis on ajoute des points avec un programme interactif. On veut lire l'écran et savoir s'il y a un point en telle ou telle position de l'écran.

Y a-t-il des PEEKs pour cela ? Comment procéder ?

Guy Fombonne - Montrouvin - Tauxigny - 37310 Reignac

En réponse à votre lettre, voici quelques précisions sur la fonction SCRN haute résolution sur Apple II. Pour tout ce qui concerne les détails techniques concernant l'organisation de la page haute résolution, je vous prie de vous reporter à mon article du numéro 5 sur HAIFA (pages 41-42), dont une des fonctions est justement le SCRN haute résolution. Si vous ne possédez pas le numéro 5, dont le second tirage est malheureusement épuisé, il vous faudra attendre la parution (très prochaine) du recueil numéro 2 de Pom's.

Eh non, il n'y a malheureusement pas de PEEK qui dit si un point de l'écran est allumé ! C'est beaucoup plus complexe, car un PEEK de l'écran donne l'état de 7 points alignés. Voir l'article de JF Duvivier - les adresses du graphique (Pom's 1 page 35 ou recueil page 24).

La meilleure méthode est d'employer un petit programme Applesoft dont →

SYBEX

Gagnez un Voyage à Silicon Valley

Oui, vous êtes invités gratuitement à gagner un voyage d'une semaine pour deux personnes au pays de la micro-informatique.

MICRO-EXPO, 9e congrès-exposition, carrefour international de la micro-informatique se tiendra à Paris, au Palais des Congrès du 22 au 26 mai 1984.

Visitez cette manifestation qui vous offrira la possibilité exceptionnelle de rencontrer et de dialoguer avec plus de 200 exposants français et étrangers, de suivre une trentaine de conférences professionnelles et grand public : comment choisir son tableur électronique, les systèmes intégrés : 1 - 2 - 3, Lisa, Visi/On, MS-WIN, choisir son micro, comptabilité et bases de données, Basic...Découvrez les dernières nouveautés dont certaines seront présentées en exclusivité.

La multiplicité et la diversité des produits et techniques présentés à ce grand rendez-vous annuel constitueront pour vous la garantie du bon investissement et de la bonne décision.

COUPON REPONSE A COMPLETER ET A RETOURNER A SYBEX
6-8, impasse du Curé - 75018 PARIS
Ce bulletin est votre titre de participation au tirage au sort qui aura lieu en présence de Maître P. Chale, huissier à Paris. Merci de m'adresser
- une entrée gratuite et le programme détaillé des conférences,
- un passeport valeur 100 F T.T.C. me donnant droit à l'entrée permanente au salon, au guide de la micro 84 et de participer à toutes les conférences de mon choix (attention le nombre de places est limité).

NOM
Prénom
Société
N° Rue
Code postal Ville
Activité de l'entreprise
Fonction
Coupon à retourner avant le 11/11/83
POM'S

MICRO EXPO

je joins le listing. Il y a une astuce qui consiste à faire deux XDRAWs successifs d'une shape qui ne comporte qu'un point. Ces XDRAWs consécutifs s'annulent, mais si l'on relève la valeur du compteur de collision de l'instruction DRAW (mémoire 234) après le second XDRAW, on trouve 1 si le point où ont été faits les XDRAWs était allumé, 0 sinon.

Bien sûr, si le programme dans lequel vous utilisez ce SCRN haute résolution possède lui-même une table de shapes, il faudra changer l'adresse de début de table (232-233) à chaque fois que vous utilisez une shape (pour indiquer soit la table du programme, soit la table du SCRN).

■

```
2   REM  "SCREEN" HAUTE-RESOLUTION

3   REM  ==========================


10  POKE 232,0: POKE 233,3: FOR I
      = 768 TO 773: READ J: POKE
     I,J: NEXT
20  DATA 1,0,4,0,4,0
30  POKE  - 16297,0: POKE  - 1630
     1,0: POKE  - 16304,0: HOME :
     VTAB 20: SCALE= 1: ROT= 0
40  PRINT : INPUT "POINT (X,Y) ?
     ";X,Y
50  XDRAW 1 AT X,Y: XDRAW 1 AT X,
     Y:SC =  PEEK (234)
60  IF SC = 1 THEN  PRINT "ALLUME
     ": GOTO 40
70  PRINT "ETEINT": GOTO 40
```

Tracé de courbes en conversationnel

Gilles Mauffrey

Ce petit programme permet d'introduire au terminal une fonction dont la variable est X et de la tracer sur la première page graphique. Par exemple :

SIN(X)
EXP(−X)*SIN(X)
(2*X)/(1−(X^2)) ...

Lorsque vous rentrez une fonction, vous devez bien sûr respecter une syntaxe compréhensible pour l'ordinateur (utilisez par exemple "3*X" et non "3X").

Pour qu'il soit possible d'introduire la fonction en conversationnel, il est nécessaire de "tokeniser" l'expression saisie au clavier : cette opération constitue l'objet du sous-programme assembleur B.TOKFON. Ce programme utilise deux routines de la ROM BASIC : le sous-programme GETLINE (en $D52C) qui correspond à la première partie de l'instruction INPUT, c'est à dire à la saisie d'une chaîne de caractères stockée à partir de l'adresse $200 et terminée par un 0. Ceci permet d'utiliser immédiatement l'autre sous-programme TOKENI ($D56D - cf POM'S 7) qui tokenise la chaîne précédemment saisie.

Cette chaîne tokenisée est alors stockée dans le programme BASIC à partir de l'adresse $080B qui correspond à l'adresse qui suit immédiatement le signe = de la définition de la fonction.

REMARQUE : si vous voulez modifier le programme n'oubliez pas de changer cette adresse ni de conserver une place suffisante pour la tokenisation de la fonction.

Le programme B.TOKFON contient une seconde partie qui permet le tracé de fonction en inverse (blanchissement de la première page graphique).

Le programme BASIC peut être agrémenté d'une "Hard-Copy", et peut également être assez facilement modifié pour réaliser le tracé de fonctions définies paramètriquement. ■

```
1   REM  FONCTION CONVERSATIONNELLE

2   REM  (APPLESOFT)
3   REM  ==-==========-=========-=====


10  DEF  FN F(X) = : REM   ATTENTI
     ON CETTE REMARQUE, TRES UTILE
      EN FAIT, DOIT CONTENIR AUTAN
     T DE TEXTE QUE POSSIBLE..CECI
      SERVIRA A METTRE LA TOKENISA
     TION DE LA FONCTION ENTREE EN
      CONVERSATIONNEL.....RECOPIER
      DONC CE TEXTE OU N'IMPORTE Q
     UOI
14 D$ =  CHR$ (4): PRINT D$"BLOAD
     8.TOKFON"
15  INVERSE : HOME : HTAB 10: VTAB
     2: PRINT "ENTRER LA FONCTION
     A TRACER": NORMAL : PRINT : PRINT
     "F(X)=";: CALL 768
20  INPUT "ABSCISSE DE DEPART :";X
     1
30  INPUT "ABSCISSE D'ARRIVEE :";X
     2
35  REM  RECHERCHE DU MINIMUM ET D
     U MAXIMUM SUR L'INTERVALLE
40 MIN =  FN F(X1):MAX =  FN F(X1)

42  PRINT : INPUT "VOULEZ VOUS LE
     GRAPHE EN INVERSE ";A$
45 I0 = 0:J0 = MIN
50  FOR I = 1 TO 280:X = X1 + (I -
     1) * (X2 - X1) / 279:Y =  FN
     F(X):
60  IF Y < MI THEN MIN = Y
70  IF Y > MAX THEN MAX = Y
80  NEXT I
81  HGR : HCOLOR= 7: IF  LEFT$ (A$
     ,1) = "O" THEN  CALL 812: HCOLOR=
     0
82  VTAB 22: PRINT "X VARIE DE ";X
     1;" A ";X2
85  PRINT : PRINT "Y VARIE DE ";MI
```

```
    N;" A ";MAX
86 J0 = 191 * (J0 - MIN) / (MAX -
    MIN): HPLOT I0,191 - J0
90  FOR I = 2 TO 279:X = X1 + (I -
    1) * (X2 - X1) / 279
100 Y =  FN F(X):J = 191 * (Y - MI
    N) / (MAX - MIN)
110  HPLOT  TO I - 1,191 - J: NEXT
    I
112  REM  TRACE DES AXES, S'ILS NE
     SORTENT PAS DE L'EPURE
115  IF MIN > 0 THEN 127
120 J0 = 191 * ( - MIN) / (MAX - M
    IN)
124 J0 =  INT (J0)
125  HPLOT 0,191 - J0 TO 279,191 -
    J0
127  IF X1 > 0 THEN 140
130 I0 = 279 * ( - X1) / (X2 - X1)
    : HPLOT I0,0 TO I0,191
135  FOR J = 1 TO 1000: NEXT
140  REM  EVENTUELLEMENT HARD COPY
     (PREMIERE PAGE EN ENTIER)
145  PRINT "VOULEZ VOIR LA FONCTIO
    N EN ENTIER :";: INPUT A$
147  IF  LEFT$ (A$,I) = "O" THEN  POKE
     - 16302,0: FOR J = 1 TO 1000
    0: NEXT : POKE  - 16301,0
150  PRINT : PRINT : PRINT : INPUT
    "VOULEZ VOUS TRACER UNE AUTRE
     FONCTION ?  ";A$
160  IF  LEFT$ (A$,1) = "O" THEN  TEXT
    : GOTO 15
170  TEXT
```

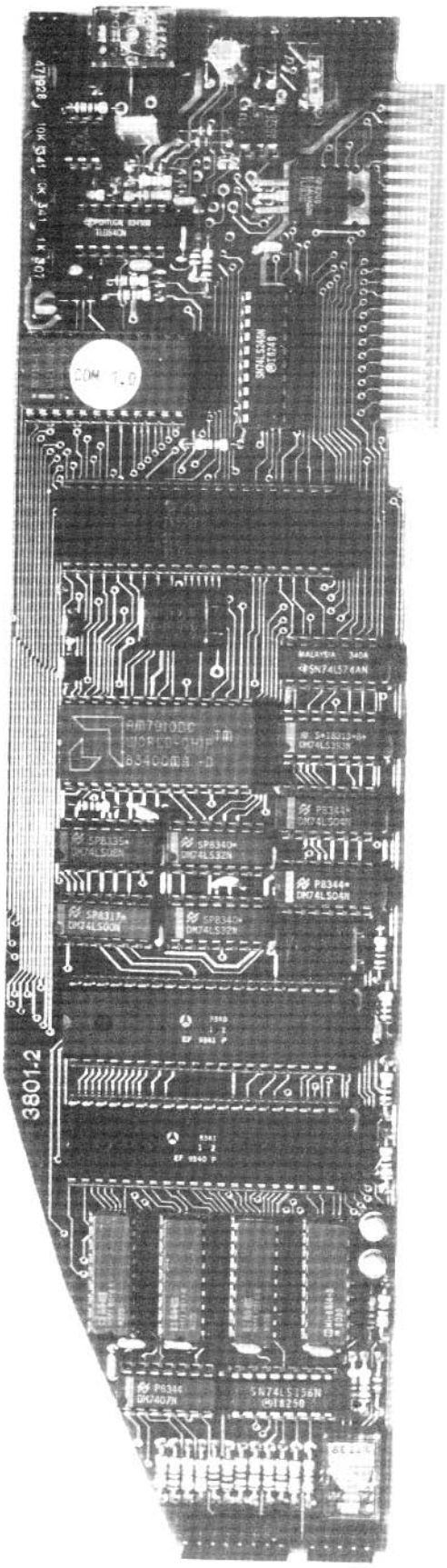
Programme BTOKFON
DOS TOOL KIT

```
0300:            1        ORG  $300
00B8:            2 TXTPTR EQU  $B8
D52C:            3 GETLIN EQU  $D52C
D56D:            4 TOKENI EQU  $D56D
080B:            5 PLACE  EQU  $80B     ;*POUR LA TOKENISATION DE LA FONCTION
0300:A5 B8       6        LDA  TXTPTR
0302:48          7        PHA
0303:A5 89       8        LDA  TXTPTR+1
0305:48          9        PHA
0306:20 2C D5 10          JSR  GETLIN
0309:A2 00       11       LDX  #00
030B:86 13       12       STX  $13
030D:20 6D D5 13          JSR  TOKENI
0310:A9 B2       14       LDA  #178     ;* TOKEN DE REM
0312:99 0B 08 15          STA  PLACE,Y
0315:88          16       DEY
0316:A9 3A       17       LDA  #$3A     ;* :
0318:99 0B 08 18          STA  PLACE,Y
031B:88          19       DEY
031C:B9 FC 01 20 TRF      LDA  $1FC,Y   ;*MISE ENPLACE DE LA TOKENISATION
031F:99 0B 08 21          STA  PLACE,Y
0322:88          22       DEY
0323:D0 F7       23       BNE  TRF
0325:68          24       PLA
0326:85 B9       25       STA  TXTPTR+1
0328:68          26       PLA
0329:85 88       27       STA  TXTPTR
032B:60          28       RTS
032C:A9 20       29       LDA  #$20
032E:85 01       30       STA  1
0330:85 02       31       STA  2
0332:A0 00       32       LDY  #00
0334:A9 00       33       LDA  #00
0336:85 00       34       STA  00
0338:A9 FF       35       LDA  #$FF
033A:91 00       36 BCL   STA  (00),Y
033C:C8          37       INY
033D:D0 FB       38       BNE  BCL
033F:E6 01       39       INC  1
0341:C6 02       40       DEC  2
0343:D0 F5       41       BNE  BCL
0345:60          42       RTS
```

RECAPITULATION

```
0300- A5 B8 48 A5 B9 48 20 2C
0308- D5 A2 00 86 13 20 6D D5
0310- A9 B2 99 0B 08 88 A9 3A
0318- 99 0B 08 88 B9 FC 01 99
0320- 0B 08 88 D0 F7 68 85 B9
0328- 68 85 B8 60 A9 20 85 01
0330- 85 02 A0 00 A9 00 85 00
0338- A9 FF 91 00 C8 D0 FB E6
0340- 01 C6 02 D0 F5 60
```

Apple-Tell est un système qui fait d'un simple Apple (votre Apple de tous les jours) un minitel intelligent.



Apple -Tell comprend :

- une carte Modem incluant un décodeur Teletel
- un logiciel d'Emulation de Terminal Minitel enrichi de trois éblouissantes fonctions (celles qui faisaient le plus défaut jusqu'à présent sur votre minitel) ;

IMPRESSION : l'imprimante *de votre Apple* est exploitée pour sortir les copies papier dont vous avez besoin lorsque vous consultez un serveur.

STOCKAGE : les disquettes *de votre Apple* sont utilisées pour enregistrer les pages dont la consultation vous est nécessaire :
- au format Teletel (c'est-à-dire telles que vous les avez reçues),
- en mode Texte pur (ASCII) pour exploitation locale ultérieure.

AUTOMATISME : l'intelligence *de votre Apple* est mobilisée pour accomplir l'interrogation automatique du serveur que vous avez désigné (appel téléphonique, orientation TRANSPAC, identification, choix successifs), enregistrer sur papier et/ou sur disque les données consultées, puis pour traiter celles-ci, en les incorporant dans votre application.
(Les Procédures d'interrogation sont créées par l'utilisateur, sans aucun langage de programmation, grâce au mode Apprentissage d'Apple-Tell.)

Evénement du dernier SICOB, salué par toute la presse, consacré Pomme d'Or 1983 par le Jury Apple, le système Apple-Tell marque une mutation décisive dans l'évolution des techniques videotex en environnement professionnel :

- point d'arrêt à la prolifération des matériels sur votre bureau (effet "mini-Sicob")
- utilisation optimale des ressources dont vous disposez déjà (disques, imprimante, logiciels, etc.),
- utilisation possible en mode Terminal autant qu'en mode Serveur (jusqu'à six portes),
- enfin (et c'est sans doute le point le plus important) JONCTION entre le monde extérieur et et les outils standard de votre Apple : l'incorporation de données dans Apple-Writer, Visicalc, Multiplan, PFS, Quick-File, etc., et même dans vos applications personnelles (comptabilité, suivi de commandes, fichier ...) devient possible.

Caractéristiques générales :
- Modem multimodes :
-1200/75, 1200 (half-duplex), 600 (half-duplex), 300 (full-duplex),
-standards CCITT et BELL (cette caractéristique unique rend accessibles les serveurs nord-américains, y compris par réseau téléphonique commuté).

- Sorties : video composite (N&B) et Peritel couleur.
- Compatible Apple 2 , Apple 2+, Apple 2e (48k, une disquette)
- Enfichable dans n'importe quel slot libre de votre Apple 2.
- Transparence totale vis-à-vis du système.

BON DE COMMANDE

HELLO Informatique - 1, rue de Metz 75010 PARIS
Tél : (1) 523 30 34 - Telex : FLASH 210500F

Nom _____ Société _____

Adresse complète _____

Ville _____ Code Postal _____ tél. _____

☐ Souhaite recevoir une documentation sur le système Apple-Tell

☐ Commande ☐ système(s) Apple-Tell au prix de F. 5900 H.T.

(règlement ci-joint par : ☐ Chèque bancaire ☐ CCP)

pom's

	Montant TTC
• **la disquette HAIFA Source** ☐ **au prix de 55 F la disquette (cf. Pom's n° 5)**	
• **le logiciel H-BASIC** ☐ **au prix de 150 F (cf. Pom's n° 8)**	
• **le logiciel MUSIC** ☐ **au prix de 80 F (cf. Pom's n° 10)**	
• **le Disk Manager** ☐ **au prix de 450 F (cf Pom's n° 11)**	
• **DBSTAG (CP/M)** ☐ **au prix de 450 F (cf Pom's n° 11)**	
• **Recueil n° 1 de Pom's** (nos 1 à 4)	
☐ avec ses 3 disquettes au prix de 270 F	
☐ sans disquette au prix de 120 F	
☐ les 3 disquettes seules au prix de 150 F	
• **Recueil n° 2 de Pom's** (nos 5 à 8) - Parution 15 avril	
☐ avec ses 4 disquettes au prix de 310 F	
☐ sans disquette au prix de 120 F	
☐ les 4 disquettes seules au prix de 190 F	
Je désire recevoir :	
• **les numéros de la revue Pom's** ☐ 4 ☐ 7 ☐ 8	
au prix de 35 F le numéro	
• **les numéros de la revue Pom's** ☐ 9 ☐ 10 ☐ 11	
au prix de 40 F le numéro	
• **les disquettes d'accompagnement des numéros**	
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11	
au prix de 55 F par disquette	
Je désire m'abonner pour 6 numéros à partir du n°	
☐ sans disquette au prix de 200 F	
☐ avec disquettes au prix de 480 F	
TOTAL :	

Envoyez ce bon de commande et votre règlement à :

Éditions MEV - 49, rue Lamartine - 78000 Versailles

Nom _____

Adresse _____

Ces tarifs comprennent l'envoi postal en France Métropolitaine, CEE et Suisse (voie aérienne exceptée)

Supplément avion : 10 F par numéro et / ou disquette

La nouvelle bibliothèque Apple

des livres utiles pour concevoir vos propres programmes sur Apple

Ecrivons un programme pour Apple

Pour vous amuser en apprenant la programmation sur un micro ordinateur qui semble doté de pouvoirs magiques. Vous apprendrez à lui donner des ordres en BASIC et, toutes sortes d'instructions. Vous pourrez vous en servir pour dessiner. Vous le comprendrez et vous saurez le faire fonctionner en ouvrant ce livre.

1C0544 **99,00 F**

Vous et l'ordinateur Apple

Un livre pour tous ceux qui veulent connaître l'APPLE et s'en faire un ami.
33 leçons pour apprendre à programmer en BASIC, chaque leçon est illustrée de commentaires, d'explications, de programmes et d'exercices complémentaires.

1C0542 **129,00 F**

Le Logo sur Apple

Un livre pour faire l'apprentissage de LOGO, un moyen d'expression très puissant. D'une façon très vivante, ce livre explique les étapes à suivre pour concevoir et réaliser toutes sortes de projets... à partir de la tortue LOGO, ce célèbre petit animal cybernétique qui se déplace sur l'écran en laissant une trace derrière lui.

1C0543 **129,00 F**

84CD6913

Jeux vidéo, jeux de demain

CEDIC/NATHAN/LEP

Les jeux vidéo et les jeux sur ordinateurs envahissent notre environnement. Pour se retrouver dans la jungle des envahisseurs, simulateurs de vol, football, joyeux voraces et autres aventures... pour mieux les comprendre et mieux les exploiter... voici des informations sur leur impact, des conseils sur la manière de les utiliser, et un banc d'essai des consoles actuelles.

1C0521 **59,00 F**

NOUVEAU

Envoi de notre catalogue sur simple demande

CEDIC/NATHAN

32, bd Saint Germain 75005 PARIS - Tél. (1) 326.42.71

LE COIN «APPLE» ET COMPATIBLE

Moniteur couleur

Moniteur carrossé pour Apple 12" ..... 3520 F
Carte RGB pour Apple II + ... 699 F

• Le moniteur idéal pour tout mini ou micro-ordinateur avec entrée RGB • Totalement compatible avec les ordinateurs individuels Apple III et IBM sans aucune interface complémentaire • Cartes interfaces «RGB II» disponibles pour compatibilité Apple II E.

Floppy disk compatibles

Strictement compatibles ces «floppy» sont garantis 1 an et commercialisés dans la version Half Size. De plus le Track to Track de 3 millisecondes les classe parmi les plus rapides 5".
Floppy sans contrôleur ..... 2699 F avec contrôleur ..... 3459 F

Joysticks pour Apple ..... 320 F

Possibilité de commuter le levier en mode stable ou instable.

Programmateur de mémoire EPROM pour APPLE

Capable de programmer les 2708, 2716, 2732, 2758, 2532. Complet tes é ..... 1562 F

Alimentation à découpage

Mêmes dimensions que l'alimentation Apple soit – 5V, 2,5 A + 12V, 1,5 A – 12V, 0,5A, – 5 V, 0,5 A ..... 799 F

Disques durs pour Apple 17500 F clés en main

6,7 Mega octets compatibles. Dos 3.3 Pascal et CP/M.

Carte RGB + 80 colonnes ..... 1640 F

pour Apple II E.

Coffret type Apple ..... 698 F

avec découpe pour pavé numérique.

Tarif Hard

APPLE II E			
Carte 80 C	998 F	Clavier numérique	1290 F
Carte 80 C + 64 K RAM	2379 F	Carte A/D 16 voies	3620 F
Carte 80 C + 64 K + Péritel	2970 F	Carte horloge	1060 F
Interface Série	1360 F	Carte 16 K Apple II +	1330 F
Interface Parallèle	1360 F	APPLE III	
Interface Modem	5700 F	Interface parallèle	1635 F
Disk Apple	2990 F	Sylentype III	2640 F
Carte PROTO	165 F	Pascal	2120 F
Poignées	280 F	Visicalc	2700 F
Z 80 avec CP/M	3100 F	Apple Writer III	1580 F
		Carte couleur Péritélévision	820 F

Compatible Apple

Cartelangage ..... 695 F
Carte Z 80 ..... 995 F
Carte 128 K RAM ..... 2200 F
Clavier type Apple ..... 950 F
Carte musicale ..... 850 F
Carte horloge ..... 785 F
Carte communication ..... 900 F
Ventilateur ..... 350 F
Carte 6522 via ..... 550 F

BIENTÔT

Une souris pour Apple II + et II E. Utilisable avec 69 K de RAM minimum et d'un lecteur avec contrôleur. Livrée avec son câble d'interface, une disquette de présentation et son guide d'utilisation.

Penta 8

34, rue de Turin, 75008 PARIS. Tél. 293.41.33.
Métro : Liège, St-Lazare, Place Clichy. Télex 614789.

Penta 13

10, bd Arago, 75013 PARIS. Tél. 336.26.05.
Métro : Gobelins (service correspondance et magasin).

Penta 16

5, rue Maurice Bourdet, 75016 PARIS. Tél. 524.23.16
(Pont de Grenelle) - Metro Charles Michels.
Bus 70/72 : Maison de l'ORTF.

Prix TTC donnés à titre indicatif pouvant varier en fonction des approvisionnements.

apple l'Ordinateur Personnel

PENTA SERVICE

PROMOTION

APPLE II E

13500 F comprenant

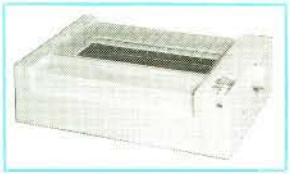
• Apple soft 12 K ROM • RAM 64 K • Clavier AZERTY (français) ou QWERTY • Alimentation à découpage • Sortie vidéo.
MONITEUR APPLE Bande passante 18 MHz • TOP lignes à 40 µs • Ecran vert • Modification de l'angle de vision • FLOPPY lecteur 5" • Capacité 143 K • Avec DOS 3,3 • Alimenté par l'unité centrale II E.

APPLE III

43800 F avec

256 K, avec disquette 5" et disque dur 5 M/octets

• UC 6502, 256 K de RAM • 4 K de ROM • Affichage 80 x 24 • Ecran vert anti-reflet • Graphisme • Couleur • Clavier 74 touches • 5 M/octets sur disque dur.

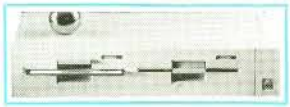


IMAGEWRITER,

l'imprimante conçue pour votre Apple

4560 F

• 180 cps • moins de 53 dBA • Matricielle 7 x 9 programmable 16 x 8 • Interface RS 232 • Mémoire tampon 1 K • Vitesse transfert 300, 1200, 2400, 9600 • Kit accessoires II ou III **272 F.**



DUO-DISK,

floppy double unité pour Apple

7580 F

• 2 x 140 K/o formatés • Kit accessoires **1135 F**

LOGICIELS APPLE II E*

(* logiciels pouvant être prêtés aux conditions du «Bon à découper» ci-dessous.)

- VISICALC (français) : traitement de planning, gestion, budget, finance, fabrication ..... 2700 F
- MULTIPLAN : Mêmes utilisations que VISICALC mais traitement d'écran plus sophistiqué ..... 2420 F
- VISIPLOT : fabrication de courbes ou d'histogrammes à partir de données numériques. Prix ..... 1630 F
- ORCA : un des meilleurs assembleur pour APPLE ..... 1490 F
- APPLE WRITER : traitement de texte 1572 F
- BUSINESS GRAPHIQUE : traitement graphique des valeurs numériques ..... 1375 F
- APPLE FORTRAN : langage ..... 1659 F
- QUICK FILE : traitement de fichier ..... 790 F
- VISIFILE : création et gestion de fichier 2833 F
- APPLE LOGO : initiation à la géométrie et à la trigonométrie ..... 1600 F
- APPLE PASCAL : langage ..... 1820 F
- CX BASE 200 : création et traitement de fichier ..... 3290 F
- CX TEXTE : traitement de texte interactif avec CX BASE 200 ..... 1190 F
- CX BASE 200 + CX TEXTE ..... 3990 F

DECOUPEZ CE BON

ASSURANCE LOGICIEL APPLE

PENTASONIC s'engage à rechercher et à démarrer avec vous parmi les logiciels standards celui qui correspond exactement à VOS BESOINS.
TESTEZ LE PENDANT 10 JOURS GRATUITEMENT

BON POUR UN ESSAI GRATUIT DE 10 JOURS

Ce bon vous donne droit au prêt d'un logiciel APPLE précédé d'un astérique dans notre publicité. A être guidé dans vos premiers pas sur ce logiciel par un technicien PENTASONIC.

Type log ..... n° de série .....

Date d'essai ..... Date de retour .....

Cette offre n'est soumise à aucune obligation d'achat. Il vous sera simplement demandé un chaque de caution.

DECOUPEZ CE BON POUR

BENEFICIER D'UN DEPANNAGE IMMEDIAT DE VOTRE MATERIEL APPLE

Attention, il est préférable de prévenir nos techniciens de votre passage!

BON DE DEPANNAGE IMMEDIAT de votre matériel APPLE ou PRET D'UN APPLE EQUIVALENT

Si le dépannage immédiat s'avère impossible*

Ce nouveau service PENTA-APPLE implique que le matériel soit d'origine PENTASONIC. Le dépannage s'effectuera dans un des points de vente PENTA.

PENTAT 8 13 6 type et n° de série .....

Date d'achat ..... N° de validation .....

* Sauf rupture de stock sur les pièces détachées d'origine.